2024

Técnicas Digitales III

Apuntes de clase

Profesor: Mg. Ing. Felipe Fernando Poblete

J.T.P.: Ing. Mariano Martín González

ÍNDICE

ÍNDICE 1

PREFACIO 9

CAPÍTULO 1 EVOLUCIÓN DE LOS MICROPROCESADORES 12

1.1 PROYECTO DE LA UNIDAD DE CONTROL 12

1.1.1 Lógica al azar 12

1.1.2 Microprogramación 12

1.1.3 Unidad de control micro programada 12

1.1.4 Comparación de ambas técnicas 16

1.2 PROCESAMIENTO PARALELO 16

1.2.1 Procesador microprogramado 16

1.2.2 Pipelining 17

1.3 PROCESADORES RISC 20

1.3.1 Procesadores actuales 20

1.4 ORGANIZACION DE MEMORIAS 21

1.4.1 Paginación 21

1.4.2 Segmentación 22

1.5 MEMORIAS CACHE 23

1.5.1 Introducción 23

1.5.2 Configuración básica 24

1.5.3 Configuración avanzada 25

1.6 SISTEMAS MULTIUSUARIO 27

1.6.1 Sistema multitarea 27

1.6.2 Sistema multiusuario 27

1.7 MODO PROTEGIDO 30

1.7.1 Microprocesador provisional 30

1.7.2 El 80286: Primer Intento de Modo Protegido 30

1.7.3 Direccionamiento Virtual 31

1.7.4 Espacios de Direccionamiento Local y Global 32

1.7.5 Manejo de Memoria Sin Preocupaciones 33

1.7.6 Protección: el Modelo de la Cebolla 33

1.7.7 Protección: el 80386 35

1.8 MEMORIA Y MÁQUINA VIRTUALES **37**
1.8.1 Memoria virtual 37
1.8.2 Máquina virtual 39
1.9 SOFTWARE PARA EJECUTAR MÁQUINAS VIRTUALES **40**
1.9.1 VirtualBox - Arquitectura 40
1.9.2 VMware - características 44
1.10 MULTIPROCESAMIENTO **46**
1.10.1 Introducción 46
1.10.2 Arquitectura Multibus 47
1.11 COPROCESADORES **52**
1.11.1 Introducción 52
1.11.2 Aplicaciones 53

CAPÍTULO 2 PROGRAMACIÓN EN AMBIENTE DE PC **54**

2.1 EL SISTEMA OPERATIVO **54**
2.1.1 Clasificación 54
2.1.2 Programación del SO 54
2.1.3 Sistema operativo DOS 55
2.1.4 Sistema de archivos 55
2.1.5 Procesador de comandos 55
2.1.6 Funciones del DOS 56
2.2 CREACIÓN DE UN PROGRAMA EN LENGUAJE ASSEMBLY **56**
2.2.1 Editor 56
2.2.2 Macro ensamblador y turbo ensamblador 57
2.2.3 Vinculación 58
2.2.4 Compilación 58
2.2.5 Depuración 59
2.3 EJEMPLOS DE PROGRAMACION EN ASSEMBLY **59**
2.3.1 Contenido del archivo ENSAYO.ASM 59
2.3.2 Contenido del archivo ORDENDES.ASM 60
2.3.3 Proceso de ensamblado y vinculación 61
2.3.4 Ejecución del programa en la línea de comando DOS 62
2.3.5 Contenido del archivo ENSAYO.LST 62
2.3.6 Contenido del archivo ENSAYO.REF 64
2.3.7 Contenido del archivo ORDENDES.LST 65
2.3.8 Contenido del archivo ORDENDES.REF 67
2.3.9 Contenido del archivo ENSAYO.MAP 67

2.4 EJEMPLOS DE PROGRAMACION EN C Y ASSEMBLY **67**
2.4.1 Contenido del archivo PRIMOS.C 67
2.4.2 Contenido del archivo CALCULA.ASM 67
2.4.3 Proceso de ensamblado, compilación y vinculación 68
2.4.4 Ejecución del programa en la línea de comando DOS 68

CAPÍTULO 3 MEDIOS DE ALMACENAMIENTO DE DATOS **70**

3.1 CLASIFICACION Y APLICACIONES **70**
3.1.1 Clasificación 70
3.1.2 Aplicaciones 71
3.2 INTERFAZ CON EL PROCESADOR **71**
3.2.1 Esquema general 71
3.2.2 Tipos de controladores/interfases 72
3.3 METODOS DE GRABACION MAGNETICA **76**
3.3.1 Introducción 76
3.3.2 Método FM (modulación de frecuencia) 80
3.3.3 Método MFM (FM modificado o Miller) 80
3.3.4 Método RLL (Run Lenght Limited) 81
3.4 ORGANIZACION Y CARACTERISTICAS **83**
3.4.1 Organización física 83
3.4.2 Formateo 83
3.4.3 Definiciones 85
3.5 UNIDAD DE DISCOS RIGIDOS **86**
3.6. UNIDAD DE DISCOS FLEXIBLES **90**
3.7. UNIDAD DE CINTA MAGNETICA **91**
3.8. UNIDAD DE DISCOS ÓPTICOS **93**
3.8.1 Unidad de CD ROM, DVD ROM y Blu-ray 93
3.8.2 Unidad RW (Read/Write) 96
3.9. ARREGLOS DE DISCOS (RAID) **97**
3.10. CONFIABILIDAD **98**

CAPÍTULO 4 ARQUITECTURA DE COMPUTADORAS PERSONALES **100**

4.1 INTRODUCCION **100**
4.2 COMPONENTES DEL SISTEMA **100**
4.2.1 Microprocesador 8088 100
4.2.2 Coprocesador matemático 8087 101
4.2.3 Controlador de interrupciones 8259 101

4.2.4 Controlador de DMA 8237 102
4.2.5 Timer 8253 102
4.2.6 Rom BIOS 102
4.2.7 BIOS Setup 103
4.2.8 Diagrama en bloques 103
4.3. BUSES EN LA ARQUITECTURA PC-COMPATIBLE 105
4.3.1 El bus ISA 105
4.3.2 El bus MCA de 32 bits 107
4.3.3 El bus EISA de 32 bits 107
4.3.4 El bus local de video o Video Local (VL) bus 109
4.3.5 El bus local PCI 110
4.3.6 El Puerto de Gráficos Acelerado (AGP) 111
4.3.7 Interfaz PCI Express (PCI-E) 112
4.3.8 El SMBus 114
4.3.9 Resumen 117
4.4 ARQUITECTURAS ACTUALES DE PC 118
4.4.1 Introducción: 118
4.4.2 Moterboard Intel 118
4.4.3 Moterboard AMD 119
4.4.4 BIOS Setup de un motherboard actual 119
4.5 MONITORES DE VIDEO 120
4.5.1 Fundamentos 120
4.5.2 Colores 120
4.5.3 Ajustes en los monitores 122
4.5.4 Tecnologías de monitores 122
4.6 TARJETAS GRÁFICAS 127
4.6.1 Fundamentos 127
4.6.2 Evolución 127
4.6.3 Interfases PC-TG 128
4.6.4 Interfases TG-Monitor 128
CAPÍTULO 5 DETECCIÓN DE FALLAS Y BENCHMARKS 130
5.1 DETECCIÓN DE FALLAS 130
5.1.1 Diagnósticos post 130
5.1.2 Diagnósticos Stand-Alone – RST, Memtest86 y WMD 131
5.1.3 Diagnósticos para DOS – PC-Check 132
5.1.4 Diagnósticos para Windows – Passmark BurnIn Test 143

5.1.5 Diagnósticos para Windows – Passmark Monitor Test 146

5.2 PROGRAMAS DE MONITOREO 147

5.2.1 Programas de monitoreo para Windows – Speedfan 147

5.2.2 Programas de monitoreo para Windows – SensorsView 150

5.2.3 Programas de monitoreo para Windows – Hardware Monitor 151

5.3 BENCHMARKS 152

5.3.1 Introducción 152

5.3.2 Benchmark para Windows – Everest 152

5.3.3 Benchmark para Windows – Sisoft Sandra 155

5.3.4 Benchmark específico para discos en Windows – CrystalDiskMark 159

5.4 PROGRAMAS DE INFORMACIÓN 160

5.4.1 Programas de información para Windows – Everest 160

CAPÍTULO 6 SEGURIDAD INFORMÁTICA 161

6.1 DEFINICIONES 161

6.2 ASPECTOS QUE ATAÑEN A LA SEGURIDAD INFORMÁTICA 161

6.2.1 Medidas de mantenimiento y prevención de intrusiones 162

6.3 VIRUS 163

6.3.1 Virus – dificultades que generan 163

6.3.2 Cosas que no son virus 163

6.3.3 Tipos diferentes de virus 166

6.3.4 Características de los virus 169

6.3.5 Daño hecho por los virus 170

6.3.6 Como se difunden los virus 171

6.4 MEDIDAS DE PREVENCION 172

6.4.1 Reglas, procedimientos, educación y herramientas 172

6.4.2 Herramientas anti-virus 173

6.4.3 Redes y virus 175

6.5 AMENAZAS ACTUALES 176

6.5.1 Spyware 176

6.5.2 Adware 176

6.5.3 Rootkits 177

6.5.4 Exploits 177

6.5.5 Gusanos 177

6.5.6 Cookies 177

6.5.7 Hoax 178

6.5.8 Virus de ejecución especulativa 178

6.5.9 Ransomware 178
6.5.10 Botnets 178
6.5.11 Phishing 178
6.5.12 Inyección de código SQL 179
6.5.13 “Man-in-the-middle” 179
6.5.14 Denegación de servicio 179
6.5.15 Backdoor 179
6.5.16 Keylogger 179
6.6 AMENAZAS INFORMATICAS DEL FUTURO 179
6.7 CIFRADO DE LA INFORMACIÓN O CRIPTOGRAFÍA 180
6.7.1 Cifrado simétrico 180
6.7.2 Cifrado asimétrico 181
6.7.3 Cifrado híbrido 182
6.8 MÓDULO DE PLATAFORMA CONFIABLE 182
CAPÍTULO 7 PROCESAMIENTO DIGITAL DE SEÑALES 184
7.1 SEÑALES 184
7.2 SISTEMAS 186
7.3 CONVOLUCIÓN 187
7.4 CORRELACIÓN 188
7.4.1 Autocorrelación 188
7.5 CONVOLUCIÓN DISCRETA 189
7.5.1 Propiedades sobre la duración de la convolución discreta. 190
7.5.2 Convolución y Correlación en MATLAB 191
7.6 SERIES Y TRANSFORMADA DE FOURIER 191
7.6.1 Series de Fourier 191
7.6.2 Efecto Gibbs 192
7.6.3 Transformada de Fourier 192
7.6.4 Relación entre series y transformada de Fourier 192
7.7 TRANSFORMADA DE LAPLACE 193
7.8 MUESTREO Y CUANTIZACIÓN 194
7.8.1 Teorema del muestreo 194
7.8.2 Cuantización 195
CAPÍTULO 8 REDES DE COMPUTADORAS 196
8.1 USOS DE LAS REDES DE COMPUTADORAS 196
8.1.1 Aplicaciones de negocios 196

8.1.2 Aplicaciones domésticas 199
8.1.3 Usuarios móviles 200
8.1.4 Temas sociales 201
8.2 HARDWARE DE REDES 202
8.2.1 Redes de área local 203
8.2.2 Redes de área metropolitana 204
8.2.3 Redes de área amplia 205
8.2.4 Redes inalámbricas 207
8.2.5 Redes domésticas 209
8.2.6 Interredes 210
8.3 SOFTWARE DE REDES 211
8.3.1 Jerarquías de protocolos 211

APÉNDICES 214

APÉNDICE A - MOTHERBOARDS DE PC 214
A.1 MOTHER BOARD DE PC-XT – Socket DIL 64 214
A.2 ARQUITECTURA DEL SISTEMA - PC-XT 215
A.3 MOTERBOARD DE PC ATX – Socket A (AMD) 216
A.4 MOTERBOARD DE PC ATX – Socket AM2 (AMD) 217
A.5 MOTERBOARD DE PC ATX – Socket AM4 (AMD) 218
A.6 MOTERBOARD DE PC ATX – Socket LGA1151 (Intel) 219
APÉNDICE B - DIRECTIVAS DEL ASM86 220
Directiva SEGMENT / ENDS 220
Directiva ASSUME 220
Directiva GROUP 221
Variables, rótulos y constantes 221
Definición e inicialización de variables 222
Operadores y expresiones 227
Operadores para forzar atributos 228
APENDICE C - TURBO DEBUG 231
APÉNDICE D – LEY DE MOORE 232
APÉNDICE E – CARACTERÍSTICAS DE LOS MICROPROCESADORES Y DISCOS RÍGIDOS 233
Tabla 1 – Microprocesadores Intel 233
Tabla 2 – Microprocesadores AMD 233
Tabla 3 – Discos rígidos 233
APÉNDICE F – ESTADÍSTICAS 234
Fig. 1 – Top ten de procesadores 234

Fig. 2 – CPU - Mejor relación Performance vs Precio 234
Fig. 4 – GPU - Mejor relación Performance vs Precio 235
Fig. 5 – Top ten de discos rígidos (incluye SSD) 236
Fig. 6 – Drives - Mejor relación Performance vs Precio 236
Fig. 7 – Top ten de memoria RAM 237
Fig. 8 – Fabricantes de memoria RAM 237
Fig. 9 – Top ten de celulares con Android 238
Fig. 10 – Fabricantes de celulares con Android 238
Fig. 11 – Top ten de supercomputadoras (Nov 2023) 239
Fig. 12 – Top ten de supercomputadoras (HPCG) (Nov 2023) 240
APÉNDICE G – LOS MICROPROCESADORES 80x86 241
G.1 EL 8086 241
G.1.1 Estructura interna 241
G.1.2 Listado de conexiones 242
G.1.3 Modelo programable 247
G.1.4 Organización de la memoria 247
G.1.5 Modos de direccionamiento 248
G.1.6 El sistema de interrupciones 250
G.1.7 Acceso directo a memoria 251
G.2 EL 80386 254
G.2.1 Estructura interna 254
G.2.2 Listado de conexiones 255
G.2.3 Modelo programable 256
G.2.4 Organización de la memoria 256
G.2.5 Modos de direccionamiento 257
G.2.6 El sistema de interrupciones 258
G.2.7. Acceso directo a memoria 258
G.3 EL PENTIUM 259
G.3.1 Estructura interna 259
BIBLIOGRAFÍA 260

PREFACIO

En 1989 no había ningún libro de texto que abarcase el programa de la materia y tampoco había material de soporte para la cátedra, por lo que las clases consistían en explicar los conceptos con el apoyo de dibujos en el pizarrón. Gracias a los alumnos del primer curso, que tomaban notas y en algunos casos grababan las clases es que pude, con la ayuda de una PC XT que tenía en mi domicilio, comenzar a escribir las páginas que verán a continuación.

Las primeras versiones estaban escritas con el procesador de textos WordPerfect y los dibujos en Dr. Genius, una especie de Paint, todo en el ambiente del sistema operativo DOS 3.3.

Este apunte de Técnicas Digitales III es la evolución y el resultado de muchos años de trabajo en la industria, en especial en el ámbito de las computadoras digitales y muchos años dedicados a la docencia terciaria y universitaria.

Vivimos una época en la que "todo está en Internet" y además tenemos Internet en el teléfono celular, por lo que se da una paradoja: ¿para qué sirve un apunte si encuentro todo inmediatamente en línea? Lo que se contrapone con ésta: de todo lo que encuentro ¿qué es lo que me sirve para Técnicas Digitales III? Bien; traté de que el apunte siga siendo lo suficientemente liviano (en MB) para que también lo pueda tener en su teléfono sin tener que descargarlo cada vez y así facilitar el acceso rápido a lo que Ud. necesita para nuestra materia.

ALCANCE

El apunte cubre todos los temas del programa de la materia, pero los aborda con extensión y profundidad diversa. Esto es porque preferí extenderme en los temas que en su momento no tenían ninguna bibliografía, siendo breve en los restantes.

La evolución de la tecnología en las últimas dos décadas ha hecho que tenga que suprimir descripciones y temas completos, esto en aras de generar el espacio para los temas nuevos y así mantener el apunte de un tamaño reducido, cumpliendo la premisa de que no sea un manual, sino un recordatorio de los temas vistos en clase.

ORGANIZACION

El apunte está organizado siguiendo los contenidos curriculares de la materia, los cuales están basados en el programa aprobado para Técnicas Digitales III. También contiene anexos, que son referenciados a lo largo del apunte, pero preferí que sean considerados como un material independiente.

LO NUEVO EN LA REVISION 2024

Esta revisión corresponde a abril de 2024.

En el capítulo 1, Se hace un reconocimiento a Maurice Wilkes. Se explica con más detalle la implementación de saltos condicionales e interrupciones externas, temas vistos anteriormente en la práctica. Se agregó la figura 1.2.6 para explicar la burbuja. Se amplió 1.3.1 con Hiper-Threading.

En la revisión de febrero de 2024. Se adecúan los números y contenidos de los capítulos a la nueva planificación curricular 2024. El Capítulo 2 sobre la arquitectura x86 pasa al

Apéndice G para mantenerlo en el apunte, pero sólo como referencia, ya que corresponde al temario de Técnicas Digitales II. Se actualizaron las estadísticas del Apéndice F a la fecha actual.

LO NUEVO EN LA REVISION 2023

En la revisión de mayo de 2023. Se explica mejor la grabación magnética perpendicular y los sensores de lectura Hall y magnetorresistivos.

En la revisión de setiembre de 2023. Se incluyen en seguridad informática los conceptos de hackeo de sensores.

LO NUEVO EN LA REVISION 2022

Se adecúa a la nueva planificación Syllabus que se puede ver en la web de la materia y en el CVG donde se puede realizar la cursada virtual. Se actualiza la carátula.

LO NUEVO EN LA REVISION 2021

Se explican mejor las interfases en el capítulo 4, se agregan imágenes de cada una y una tabla comparativa de las características. Se explican mejor los métodos de grabación. Se explica mejor y con más figuras la unidad óptica. Se reorganizó el capítulo 7 de Seguridad informática, se actualizó con las amenazas actuales y se agregaron definiciones y conceptos. Se agregaron anexos actualizados, incluyendo uno de Supercomputadoras.

LO NUEVO EN LA REVISION 2020

Dada la situación que generó la Pandemia, en la UTN FRSN se desarrollan las clases sincrónicas virtuales. Las modificaciones principales en este apunte se refieren a figuras y explicaciones más detalladas, que facilitan el cursado en esta modalidad. Se incorporaron más presentaciones en PPT y animaciones, que se pueden ver en la página web de Técnicas digitales III en https://www.frsn.utn.edu.ar/tecnicas3.

LO NUEVO EN LA REVISION 2019

Se actualizaron algunas tablas y características de componentes para adecuarlos a la fecha de esta revisión. Se incorporó la BIOS con interfaz UEFI. Se agregaron los virus de ejecución especulativa.

LO NUEVO EN LA REVISION 2016

Este Prefacio es la primera vez que lo escribo. Me pareció conveniente para recordar los orígenes de este apunte, agradecer, hacer explícito el copyright y a partir de ahora, a mencionar los temas modificados, suprimidos o ampliados.

En el capítulo 4 se incorporó una breve explicación de los cálculos de confiabilidad que aplican a los conjuntos RAID y a otros equipos usados en la industria.

DERECHOS DE AUTOR (COPYRIGHT)

Las primeras versiones de este apunte estuvieron en la fotocopiadora de la facultad desde 1992 y la versión digital está en Internet desde 1998. Aunque su uso está destinado a los alumnos de nuestra cátedra, ni nosotros ni la UTN FRSN tenemos inconveniente en que el material que contiene sea utilizado libremente. Algunas partes están tomadas de manuales o sitios web, en esos casos se cita la fuente al pie del artículo. Se ha realizado el mayor esfuerzo para evitar los errores, no obstante, hacemos la salvedad de que su exactitud no está asegurada. Por otra parte, le solicitamos cordialmente que, si lo utiliza, total o parcialmente, cite las fuentes y autores.

RECONOCIMIENTOS

A los alumnos del primer curso, en 1989, por las notas y grabaciones.

A los auxiliares de cátedra, Guillermo Tomassini, Adrián Riolfo, Diego Capucchio y Sebastián Cerella, por sus aportes a este apunte.

A Mauricio Caggioli por la introducción a los DSP, en 2003.

A Jordán Tello por la primera versión web de los apuntes, el "arte" de la carátula original de 1998 a 2021 y su apoyo permanente en el soporte de la página web que reside en los servidores de la UTN FRSN.

A Félix Sesma, mi jefe de cátedra en 1982 en la UNR, por su ejemplo de dedicación a la docencia y de preparación de material didáctico. Él fue quien me ofreció mi primer trabajo rentado como docente, minutos después de haber obtenido el título de ingeniero, en 1981.

A Oscar Luzuriaga, por brindarme con generosidad sus conocimientos en los primeros años de mi trabajo en la industria.

A Andy Tanenbaum, por su extensa y completa bibliografía, que desde hace años nos facilita el entendimiento de nuestra materia.

A Somisa, Aceros Paraná, Siderar, Ternium y empresas clientes y proveedores, sus técnicos, ingenieros y directivos, por la formación constante en el país y en el exterior, lo que me permitió estar siempre actualizado en la profesión.

Al JTP Mariano Martín González, por sus importantes aportes a este apunte, a su compromiso con la cátedra y con la docencia, su perseverancia en la formación permanente y a su calidad como persona.

Por último, a mi familia, Elvira, Sofía e Ignacio que, comprendiendo mi vocación, han cedido muchos fines de semana y algunos días de vacaciones para la preparación de los apuntes, las presentaciones, la página web, los simuladores y el resto del abundante material de la asignatura.

Espero que le sea de utilidad.

Prof. Mg. Ing. Felipe Fernando Poblete, San Nicolás, abril de 2024.

CAPÍTULO 1

EVOLUCIÓN DE LOS MICROPROCESADORES

1.1 PROYECTO DE LA UNIDAD DE CONTROL

1.1.1 Lógica al azar

A mediados de los años 70', la técnica preponderante en el diseño de los microprocesadores de 8 bits era la de "lógica al azar". Esta consistía en definir la arquitectura del procesador, lo cual incluye cantidad, tamaño y función de los registros, el conjunto de instrucciones y los servicios de entrada/salida. Luego, aplicando el álgebra de circuitos lógicos y las técnicas de diseño de circuitos combinacionales y secuenciales y algoritmos de minimización se obtenía el circuito electrónico que cumplía las especificaciones.

1.1.2 Microprogramación

La microprogramación, fue desarrollada por Maurice Wilkes en la década de 1950 mientras trabajaba en la Universidad de Cambridge. La primera implementación de la unidad de control microprogramada se realizó en el EDSAC 2.

El concepto de control microprogramado se popularizó rápidamente y se convirtió en un elemento fundamental en el diseño de muchos sistemas informáticos posteriores. Varias empresas adoptaron esta tecnología en sus computadoras, pero una de las primeras en hacerlo fue IBM, quien la utilizó en sus computadoras System/360, lanzadas en la década de 1960, lo que permitió una mayor flexibilidad y capacidad de actualización en comparación con los sistemas informáticos anteriores. Digital Equipment Corporation incorporó en 1980 a Wilkes a su equipo de ingeniería a los 67 años.

La microprogramación consiste en descomponer las instrucciones en operaciones elementales, llamadas microinstrucciones e implementar el hardware partiendo de una base común (véase la figura 1.1.1) donde todas las instrucciones utilizan los mismos circuitos.

1.1.3 Unidad de control micro programada

Se analizará el diagrama esquemático de la figura 1.1.1, que muestra una unidad de control micro programada, basada en un diseño de Digital Equipment Corporation (DEC PDP11/44). La línea de puntos divide el procesador en dos bloques con funciones bien diferenciadas: la lógica de control y el camino de los datos.

Lógica de control: está compuesta por la memoria de microprograma, el secuenciador y el registro de micro palabra.

Memoria de microprograma: es una memoria de alta velocidad que contiene la secuencia de operaciones elementales que componen cada instrucción. Es el núcleo del procesador, ya que de su contenido depende como responderá el mismo a los distintos códigos de instrucciones. La cantidad de locaciones oscila entre 500 y 4000 y el ancho de palabra entre 24 y 100 bits.

Secuenciador: es el encargado de direccionar la memoria de microprograma (control store) basándose en el contador de microprograma MPC (Micro Program Counter). El MPC es un registro que se incrementa con un reloj interno, pero que depende de los registros de estado e instrucción y de algunos bits del registro de micro palabra. Ante determinadas situaciones toman valores preestablecidos (RESET, INTERRUPT, HOLD, etc.)

Registro de estado: proviene de la unidad aritmética y lógica y refleja el estado (status) de los indicadores (flags) luego realizar una operación (CARRY, ZERO, OVERFLOW, etc.). Su contenido es tenido en cuenta por el secuenciador cuando la instrucción en curso es un salto condicional.

Registro de instrucción: es el código de operación de la instrucción actual. Interviene en la fijación del valor del contador de microprograma dependiendo de la Decodificación de la instrucción y del modo de direccionamiento.

Registro de micro palabra: retiene la microinstrucción que el secuenciador direccionó sobre la memoria de microprograma y asigna bits en posiciones preestablecidas para dos objetivos principales: el control del "camino de los datos" (data path) para habilitar multiplexores, buffers y seleccionar la operación lógica en la ALU y por otra parte el control del secuenciador en función de los bits de micro bifurcación, que intervienen en las instrucciones de salto condicional e interrupciones.

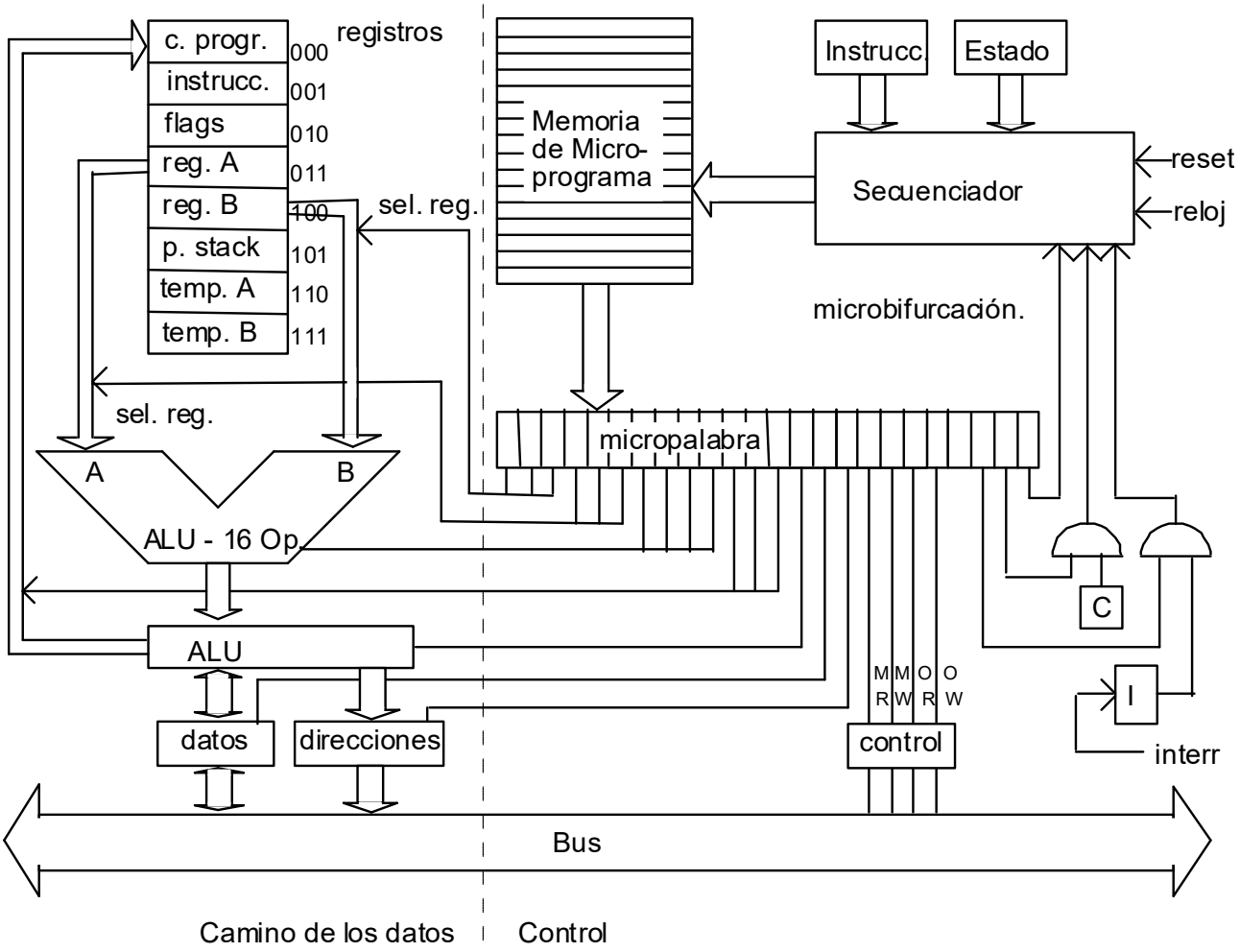


Fig. 1.1.1 - Unidad de control microprogramada

Camino de los datos: está compuesto por los registros de uso general, la unidad aritmética y lógica y la interfaz al bus.

Registros de uso general: incluyen los registros del modelo programable (los que ve el programador en assembly) y otros internos que intervienen como temporarios (temporary register) en operaciones más complejas, como por ejemplo la multiplicación.

Unidad aritmética y lógica: en ella reside la capacidad de procesador para efectuar operaciones lógicas elementales (AND, OR, INCREMENTO, COMPLEMENTO (/), NEGACION (-), etc.) sobre los datos que los multiplexores de registros colocan en cada una de sus "piernas", almacenando el resultado en otro registro. La cantidad de bits que la ALU puede manejar a la vez determina el ancho de palabra de la unidad de control que se está diseñando, por ejemplo, una ALU de 16 bits podrá sumar dos datos de 16 bits cada uno.

Interfaz al bus: con este registro, habilitado por la micro palabra, se representa en forma esquemática la interfaz entre el procesador y el mundo exterior y a partir de aquí se formarán los buses de datos, direcciones y control.

Para controlar el camino de los datos, los bits de la micro palabra están agrupados como indica la Tabla 1.1.1.

Dirección	*ALUB*	*ALUA*	*ALUFUN*	*ALUR*	*ALUCTL*	*BUSCTL*	*TEST*	*Total*
Bits	3	3	4	3	3	4	4	**24**

Tabla 1.1.1 – Organización de la memoria de microprograma (control store)

ALUB agrupa los 3 bits que seleccionan, a través de un multiplexor, el registro que se enviará a la rama B de la ALU. De la misma forma **ALUA** lo hace para la rama A.

ALUFUN selecciona la función lógica, según la codificación que muestra la Tabla 1.1.2.

ALUFUN	*Función*	*ALUFUN*	*Función*	*ALUFUN*	*Función*	*ALUFUN*	*Función*
0000	NOP	0100	A + B	1000	A XOR B	1100	A AND B
0001	A	0101	B	1001	- A	1101	A < B
0010	A=B	0110	A+1	1010	A-1	1110	SH IZQ A
0011	/A	0111	A-B	1011	A OR B	1111	SH DER A

Tabla 1.1.2 – Funciones de la ALU

ALUR controla el multiplexor que envía el resultado de la ALU o el registro de datos hacia uno de los 8 registros de uso general.

ALUCTL posee un bit que, estando en 0 habilita la escritura hacia arriba, al registro determinado por ALUR. En cambio, en 1 lo hace hacia abajo, al registro de datos. Los otros dos bits controlan la habilitación de los registros de datos y direcciones respectivamente.

BUSCTL agrupa los cuatro bits que habilitan la escritura o lectura de posiciones de memoria o E/S externas.

TEST son los bits de test para las bifurcaciones condicionadas por eventos externos (ej. Interrupciones), internos (ej. Excepciones), saltos condicionales o para indicar al secuenciador el fin de una secuencia de microinstrucciones, cuando el último bit está en 1.

Nótese que, ante una interrupción externa, aún en el caso en que el bit de habilitación de la palabra de estado lo permita, el microprograma, es decir, la secuencia de microinstrucciones que implementan una instrucción no será interrumpido. Por el contrario, la existencia o no de una interrupción externa será tratada como otro caso de bifurcación condicional mediante la selección de los bits apropiados del campo “Test”.

En los casos mencionados, la señal enviada al secuenciador tendrá dos estados posibles: **activado**, en el cual se modifica el contador de microprograma para que realice las operaciones

(ejecutar el salto condicional o la interrupción) y **desactivado**, en el que el contador no es afectado y por tanto se continúa la secuencia de instrucciones.

Ejemplo de microprogramación – Instrucción INC A:

La dirección de memoria 82h contiene 02h, que corresponde a incrementar el reg. A

El valor inicial del reg. A es 20h y el PC contiene 82h.

El microprograma debe hacer lo siguiente:

1- Decodificarla y ejecutarla (la instrucción fue traída por la instrucción previa)

2- Incrementar el PC

3- Traer la próxima instrucción de la memoria

Para eso, hay que armar una tabla que representa el contenido de la memoria de microprograma (MP):

Dir. MP	*ALU B*	*ALU A*	*ALU FUN*	*ALU R*	*ALU CTL*	*BUS CTL*	*TEST*	*Comentario*	*PC*	*ALU*	*DIR*	*A*
(*) 20	xxx	011	0110	111	000	0000	0000	Inc A y almac. en Temp B	82	21	82	20
21	xxx	111	0001	011	000	0000	0000	Almacena temp B en reg. A	82	21	82	21
22	xxx	000	0110	111	000	0000	0000	Inc PC y pone en temp B	82	82	82	21
23	xxx	111	0001	000	000	0000	0000	Poner temp B en PC	83	83	82	21
24	xxx	000	0001	xxx	101	0000	0000	Poner PC en bus direcc	83	83	83	21
25	xxx	000	0001	001	000	1000	0000	Lee bus de datos y pone en reg Instrucción				
26	xxx	xxx	0000	111	000	0000	0001	Decodifica la próxima instr. Fin de secuencia	83	xx	83	21

Tabla 1.1.3 – Ejemplo de microprogramación

(*) Cuando 02 pasa al registro de instrucción, el contador de microprograma pasa a contener 020, es decir la primera microinstrucción de la secuencia que incrementa A.

Para la realización de los trabajos prácticos de microprogramación, se desarrolló el simulador **UC1-TD3**, que se puede descargar de la página web de la materia. El mismo se implementó mediante un software de diseño de circuitos digitales, llamado Multimedia Logic

1.6.1, de Softronics Inc. El objetivo de diseño del circuito fue respetar lo más fielmente posible el funcionamiento de la unidad de control microprogramada explicada en clase.

1.1.4 Comparación de ambas técnicas

El gráfico de la figura 1.1.2, representa en abscisas la complejidad de un circuito (cantidad de instrucciones, modos de direccionamiento, ancho de palabra, etc.) y en ordenadas el costo de desarrollo y producción.

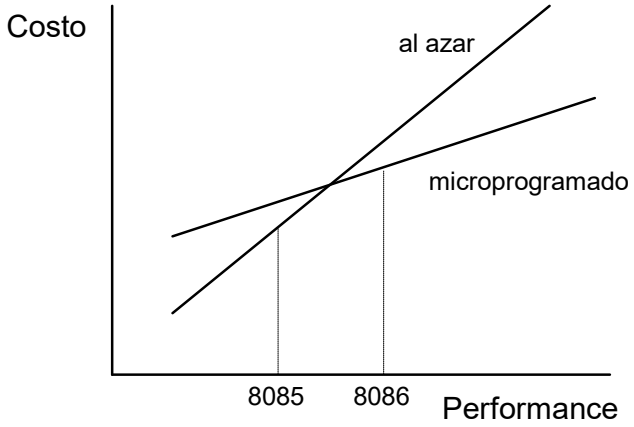


Fig. 1.1.2 – Comparación de ambas técnicas

Se observa que la pendiente que corresponde al micro programado es más suave, por lo tanto, a partir de cierta complejidad es más conveniente el diseño con esta técnica. Como ejemplo se indican los microprocesadores 8085 y 8086 de Intel.

1.2 PROCESAMIENTO PARALELO

1.2.1 Procesador microprogramado

En un procesador microprogramado genérico como el descripto en 1.1.3., la secuencia de operaciones de una instrucción típica puede representarse como sigue:

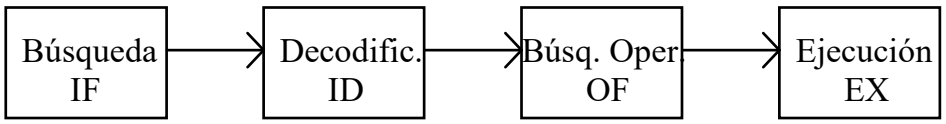


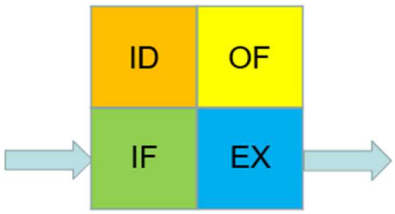
Fig. 1.2.1 – secuencia de operaciones

En un microprocesador de segunda generación como el 8085 de Intel, esta secuencia, expresada en función del tiempo y para un programa que contiene la sucesión de instrucciones I1, I2, I3, se vería así:

EX	.	.	.	I1	.	.	.	I2	.	.	.	I3
OF	.	.	I1	.	.	.	I2	.	.	.	I3	.
ID	.	I1	.	.	.	I2	.	.	.	I3	.	.
IF	I1	.	.	.	I2	.	.	.	I3	.	.	.
T	1	2	3	4	5	6	7	8	9	10	11	12

Fig. 1.2.2 – secuencia de operaciones – diagrama temporal

Si se diseña un procesador en el cual los circuitos que realizan la búsqueda de instrucciones (IF), la Decodificación de instrucciones (ID), la búsqueda de operandos (OF) y la ejecución (EX) no tienen elementos en común, es decir que pueden operar independientemente uno de otro, se obtiene un procesador paralelo de cuatro unidades.



1.2.2 Pipelining

El diagrama de tiempos, para la misma secuencia de instrucciones del caso precedente es como se muestra:

Esta técnica se denomina también "pipelining" porque asocia las cuatro fases de la ejecución de instrucciones a cuatro "caños" por donde fluyen las fases de cada instrucción.

EX	.	.	.	I1	I2	I3	I4	I5	I6	I7	I8	I9
OF	.	.	I1	I2	I3	I4	I5	I6	I7	I8	I9	.
ID	.	I1	I2	I3	I4	I5	I6	I7	I8	I9	.	.
IF	I1	I2	I3	I4	I5	I6	I7	I8	I9	.	.	.
T	1	2	3	4	5	6	7	8	9	10	11	12

Fig. 1.2.3 – secuencia de operaciones con pipeline – diagrama temporal

Cañería de ejecución
Cañería de búsqueda de operandos
Cañería de Decodificación de instrucción
Cañería de búsqueda de instrucción

Fig. 1.2.4 – pipeline de 4 etapas

La mejora en el tiempo de procesamiento, una vez que se llena la cañería, se calcula así:

El tiempo para ejecutar n instrucciones es:

Tk = k+n–1 (procesador paralelo de k unidades)

Ts = n.k (procesador tradicional)

Speedup = Ts / Tk = n.k / (k+n-1) (mejora en el tiempo)

Como n >> k … **Speedup ≈ k**

La mejora está dada por la cantidad de unidades que operan en paralelo.

K	.	.	.	I1	.	.	.	.	.	.	.	In
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1	I1	.	.	Ik	.	.	.	.	.	.	.	.
T	1	.	.	.	.	.	.	.	.	.	.	k

Fig. 1.2.5 – mejora en la performance

Este análisis simplificado considera que los tiempos involucrados en cada una de las operaciones son los mismos, cosa que en un procesador tradicional no es exactamente cierta. Los tiempos de espera introducidos degradan la performance alejándola de la ideal.

Otro factor indeseable son las rupturas de secuencia (saltos) que invalidan todas las instrucciones buscadas con anticipación y obligan a reiniciar el ciclo de llenado de la cañería.

En la figura siguiente consideramos que I2 corresponde a la ejecución de un salto a I10, entonces todas las actividades resaltadas en rojo no son de utilidad y las resaltadas en azul serán canceladas. El hueco que se produce por esta situación se conoce como burbuja.

EX	.	.	.	I1	I2	I3	I4	I5	I10	I11	I12	I13
OF	.	.	I1	I2	I3	I4	I5	I10	I11	I12	I13	.
ID	.	I1	I2	I3	I4	I5	I10	I11	I12	I13	.	.
IF	I1	I2	I3	I4	I5	I10	I11	I12	I13	.	.	.
T	1	2	3	4	5	6	7	8	9	10	11	12

Fig. 1.2.6 – problema de la burbuja

En el procesador 80486 de Intel se distinguían 5 operaciones:

WB	.	.	.	.	I1	I2	I3	I4	I5	I6	I7	I8
EX	.	.	.	I1	I2	I3	I4	I5	I6	I7	I8	.
AG	.	.	I1	I2	I3	I4	I5	I6	I7	I8	.	.
ID	.	I1	I2	I3	I4	I5	I6	I7	I8	.	.	.
IF	I1	I2	I3	I4	I5	I6	I7	I8	.	.	.	.
T	1	2	3	4	5	6	7	8	9	10	11	12

Fig. 1.2.7 – secuencia de operaciones en el procesador 80486

Donde:

IF: Instruction fetch – Búsqueda de instrucción

ID: Instruction decode – de instrucción

AG: Address generation - Generación de direcciones

EX: Execution - Ejecución

WB: Write back – Escritura en la caché interna

En el procesador Pentium se incluyó una segunda unidad de ejecución, por lo que pudo manejar en paralelo dos instrucciones, quedando el diagrama de la figura 1.2.7.

Donde:

IF1: Instruction fetch del procesador 1

IF2: Instruction fetch del procesador 2

WB2	.	.	.	.	I2	I4	I6	I8	I10	I12	I14	I16
EX2	.	.	.	I2	I4	I6	I8	I10	I12	I14	I16	.
AG2	.	.	I2	I4	I6	I8	I10	I12	I14	I16	.	.
ID2	.	I2	I4	I6	I8	I10	I12	I14	I16	.	.	.
IF2	I2	I4	I6	I8	I10	I12	I14	I16	.	.	.	.
WB1	.	.	.	.	I1	I3	I5	I7	I9	I11	I13	I15
EX1	.	.	.	I1	I3	I5	I7	I9	I11	I13	I15	.
AG1	.	.	I1	I3	I5	I7	I9	I11	I13	I15	.	.
ID1	.	I1	I3	I5	I7	I9	I11	I13	I15	.	.	.
IF1	I1	I3	I5	I7	I9	I11	I13	I15	.	.	.	.
T	1	2	3	4	5	6	7	8	9	10	11	12

Fig. 1.2.8 – secuencia de operaciones en el procesador Pentium

1.3 PROCESADORES RISC

Los procesadores RISC (Reduced Instruction Set Computer, o Computadora de Conjunto de Instrucciones Reducido) fueron diseñados para sacar provecho del procesamiento paralelo.

El conjunto de instrucciones se redujo a las operaciones elementales, con el objeto de que cada una de ellas pueda ejecutarse en un solo ciclo de reloj.

Dado que este diseño requiere menos transistores que un procesador tradicional, la superficie de chip que sobra se destina a registros (poseen muchos) y a memoria de alta velocidad (caché). Esto reduce la necesidad de acceder al bus para buscar operandos y registros intermedios aumentando la performance.

Como desventaja, puede citarse que los programas (el código objeto más precisamente) requieren más espacio por tener instrucciones menos potentes. En cuanto a los compiladores, se exige que sean optimizados para el procesador en cuestión.

Para evitar el problema de las burbujas, es conveniente el uso de lenguajes de programación estructurados, que reducen la cantidad de saltos.

1.3.1 Procesadores actuales

Debido a que en la actualidad es posible integrar miles de millones de transistores en el mismo chip, los pros de ambas técnicas se pueden aprovechar para lograr procesadores RISC con alto paralelismo, que también ejecutan instrucciones CISC, las que favorecen la programación y son compatibles con las CPU de generación anterior.

Predicción de trayectoria

A partir del Pentium, los procesadores tienen una función denominada “predicción de trayectoria”. La misma consiste en establecer un “mapa” de los lugares a donde es posible que se dirija la instrucción para adelantar la ejecución próxima. El método se basa en un buffer de decisión de destino (BTB), que incluye tres elementos por cada entrada: la dirección de la instrucción de salto, la dirección de destino de la instrucción y los bits de historia. Se usa una tabla de hasta 256 entradas para predecir los resultados de las decisiones. Los bits de historia indican si la decisión se tomó o no. Esta mejora tiende a evitar las burbujas que se producen ante saltos condicionales e Intel declara que esta característica incrementa el rendimiento en un 25%.

Hyper-Threading

Es una marca registrada de la empresa Intel para promover la implementación de la tecnología multihilo simultáneo, también conocido como SMT. Permite que los programas que estén preparados para ello ejecuten tareas usando múltiples hilos, lo cual es un procesamiento en paralelo dentro de un único procesador, incrementando así el uso de las unidades de ejecución (ALUs, FPs, etc.) del procesador.

La tecnología Hyper-Threading consiste en simular dos procesadores lógicos dentro de un único procesador físico. El resultado es una mejora en el rendimiento del procesador, puesto que al simular dos procesadores se pueden aprovechar mejor las unidades de cálculo manteniéndolas ocupadas durante un porcentaje mayor de tiempo. Esto conlleva una mejora en la velocidad de las aplicaciones que según Intel es aproximadamente de un 60 %.

El equivalente de AMD se denomina SMT.

Ref: Hyper-threading - Wikipedia

En el apéndice E se detallan algunas de las características principales de los procesadores de Intel, AMD e IBM, desde los primeros chips hasta la actualidad.

1.4 ORGANIZACION DE MEMORIAS

1.4.1 Paginación

En este ejemplo, se explicará la técnica que utilizaban los procesadores DIGITAL PDP-11 para direccionar más de 64 kB con registros de 16 bits.

Los tres primeros bits de la dirección virtual se utilizan para determinar cuál será el registro de página (PAR) seleccionado, el cual se sumará con un desplazamiento de seis bits a la dirección virtual para obtener un resultado de 22 bits (dirección física).

De esta forma, un procesador que sólo contaba con registros de 16 bits podía direccionar 4 MB de memoria física, aunque solamente 64 kB a la vez, en "páginas" de 8 kB.

Cambiando el contenido de un PAR se podía acceder a otra parte de la memoria física y así se facilitaba la conmutación de tareas en sistemas multitarea.

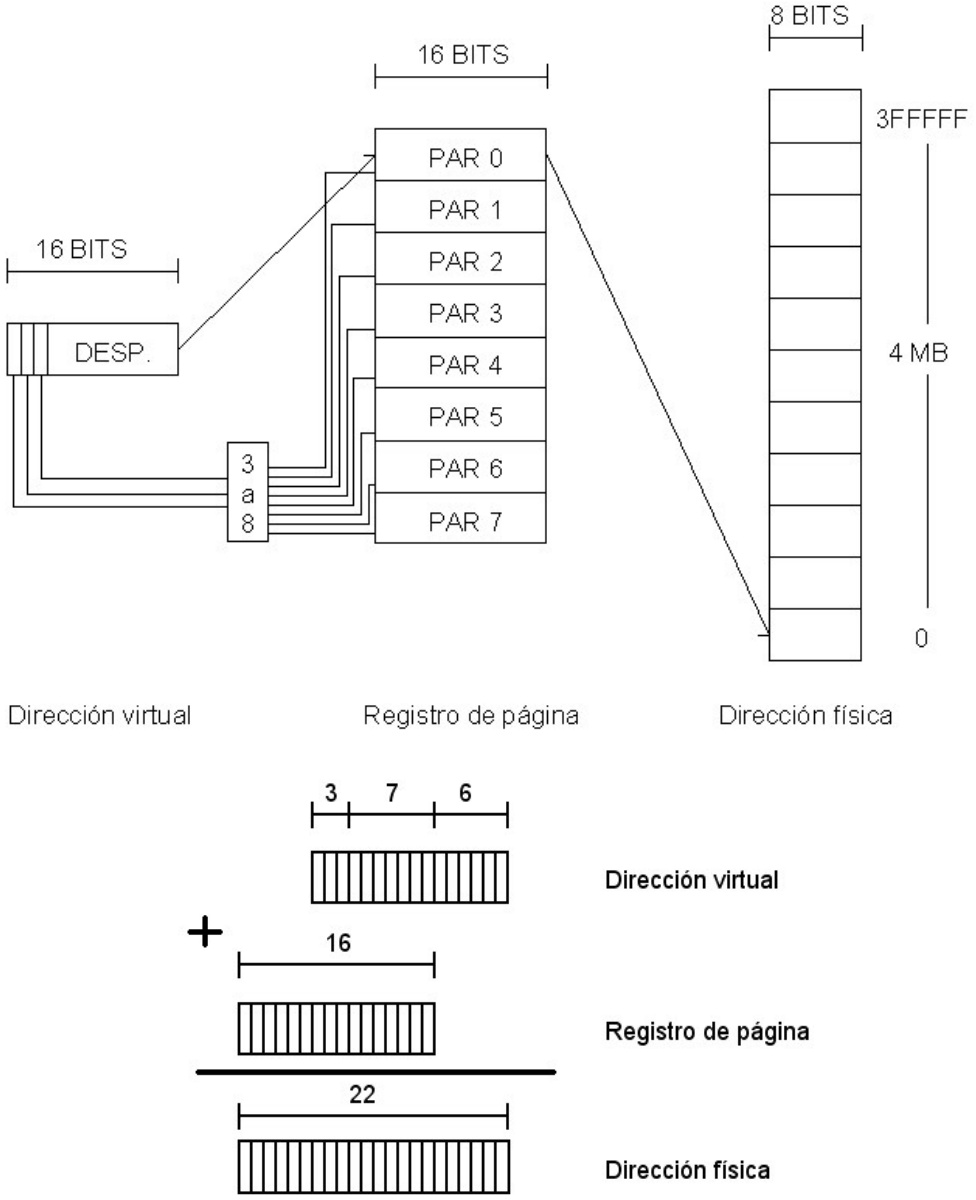


Fig. 1.4.1 - Paginación

1.4.2 Segmentación

En este ejemplo, se explicará la técnica que utilizaban los procesadores Intel 8086/80286 para direccionar más de 64 kB con registros de 16 bits.

Los registros de segmento eran CS (code), DS (data), ES (extra) y SS (stack).

La forma de sumar los registros para obtener la dirección física era la misma que en el caso de la paginación. La diferencia aquí es el modo de seleccionar el registro de segmento, que consiste en determinar el tipo de operación a realizar. Por ejemplo, si se trata de una instrucción MOV, se usará el registro DS, si es un JMP, el registro CS, si es PUSH, el registro SS.

Con un prefijo, el 8086 podía forzar un registro en particular.

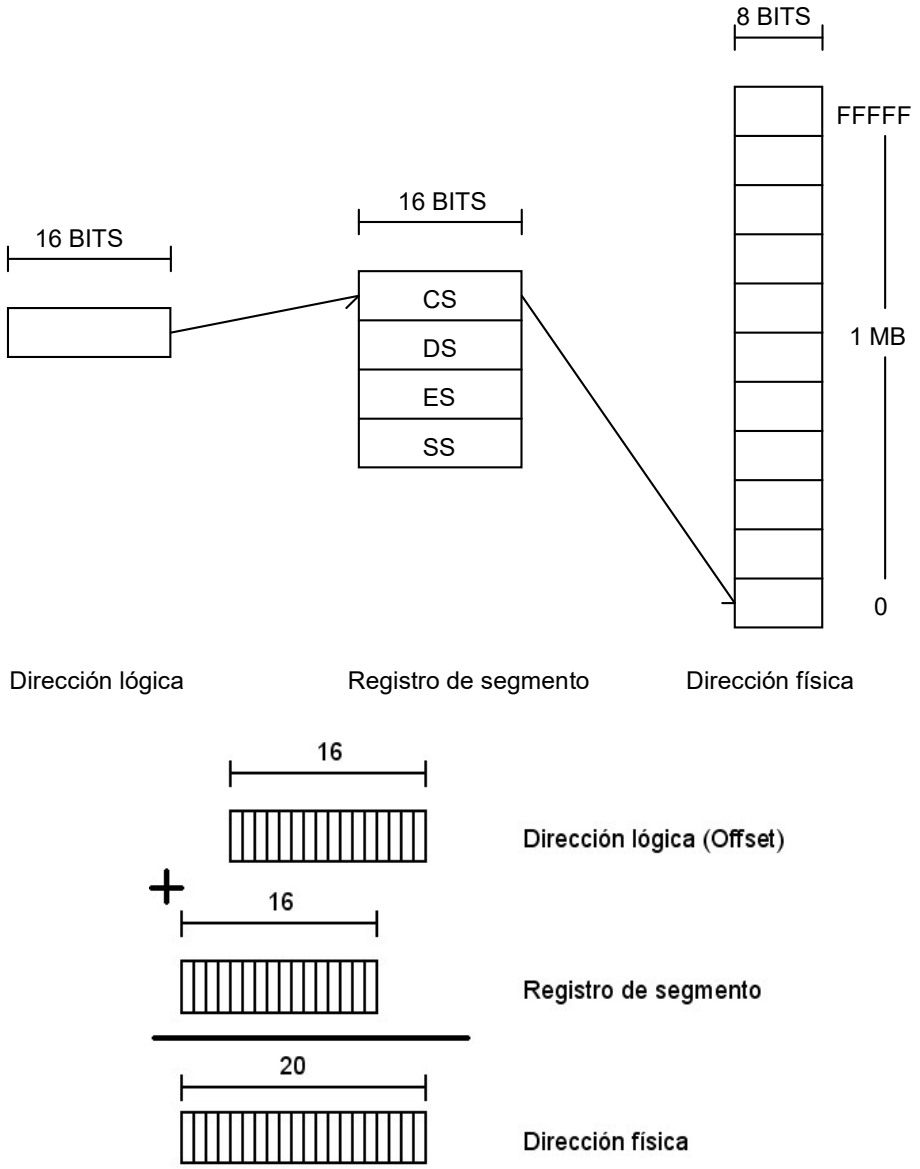


Fig. 1.4.2 - Segmentación

Aplicando segmentación (S), paginación (P) o ambas se logra:

- Aumentar el espacio de direcciones físico con registros existentes (S)(P)
- Aprovechar el espacio de direcciones virtual e implementar memoria virtual (P)

- Establecer protecciones a espacios de memoria (P)
- Aumentar el rendimiento de la cache separando en código y datos (S)
- Ejecutar código compartido entre programas (S)(P)
- Ejecutar rutinas reentrantes con distintos datos (S)(P)
- Reubicar código con facilidad (S)
- Favorecer el cambio de contexto en sistemas multitarea (S)(P)

1.5 MEMORIAS CACHE

1.5.1 Introducción

La demanda permanente de performance en las computadoras exigió siempre mejoras en cada uno de sus componentes: procesador, memoria y entrada/salida. En lo que atañe a las memorias, la reducción del precio por byte y la creciente densidad de integración permitieron aumentar la complejidad de los programas, pero como contrapartida, se requería mayor velocidad de ejecución para mantener la interactividad con el usuario.

El tiempo que demandan la búsqueda de instrucciones, operandos y el almacenamiento de resultados en memoria dependen esencialmente del *tiempo de acceso* de las memorias. A mediados de la década de los 70', existían las memorias bipolares, de baja densidad de integración, alta disipación, alto costo por bit y bajo tiempo de acceso. Estas memorias eran utilizadas principalmente para registros. También existían las memorias MOS, de mayor densidad de integración, baja disipación, bajo costo por bit y elevado tiempo de acceso.

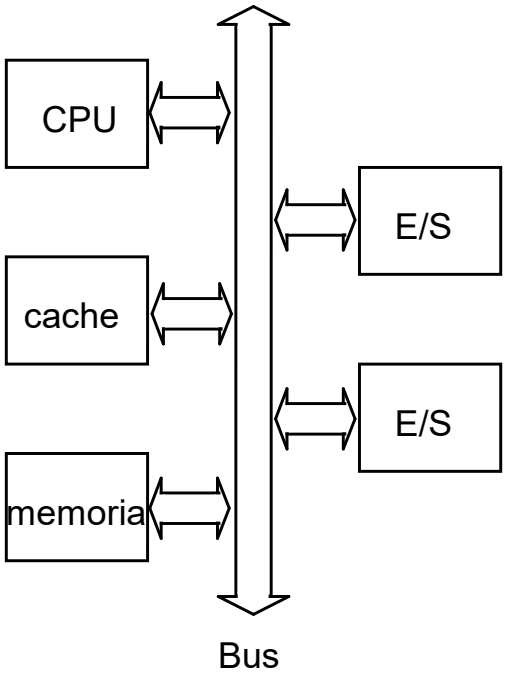


Fig. 1.5.1 - Diagrama en bloques

Esta situación obligó a los diseñadores de computadoras, ya en la década de los 70, a pensar en algo para resolver la disyuntiva: memoria veloz, pero de elevado costo (bipolar) o baja velocidad con mayor capacidad (MOS). En la figura 1.5.1 se puede observar la disposición circuital original de la memoria cache.

Las características de esta memoria eran las siguientes: tecnología bipolar (estática), capacidad 8 kB y tiempo de acceso 70 ns. La memoria convencional, de tecnología MOS (dinámica) tenía una capacidad de 4 MB y tiempo de acceso 600 ns.

1.5.2 Configuración básica

El circuito opera de la siguiente forma. Tanto la memoria cache como la convencional están conectadas al bus del sistema y por lo tanto ambas reciben la dirección solicitada por la CPU. Esta configuración se denomina: memoria curiosa.

Al principio, cuando la memoria cache está vacía, el dato pedido por el procesador será suministrado por la memoria convencional, en 600 ns, pero la memoria cache guarda tanto el dato como la dirección y un bit adicional que indica la validez del dato (véase la organización interna de la memoria cache en la figura 1.5.2).

Valid.	Dirección	Dato

Fig. 1.5.2 - Organización interna

La próxima vez que la CPU requiera el contenido de esa posición de memoria, la memoria cache podrá suministrarlo en 70 ns. Si se da el caso, como frecuentemente ocurre durante la ejecución de un programa real, que el acceso a esa posición de memoria hace falta muchas veces más, por ejemplo, en una iteración, todas las lecturas posteriores demandarán 70 ns. Si las iteraciones fueron 100, el tiempo promedio será:

$$Tp = (600 + 70 \times 99) / 100 = 75 \text{ ns}$$

Y la mejora en el tiempo de ejecución será:

$$Sp = 600 \text{ ns} / 75 \text{ ns} = 8$$

En estas condiciones la computadora se desempeña con una velocidad 8 veces mayor, casi como si toda su memoria fuese de 70 ns. Esto es válido sólo si el programa que itera cabe completamente en la memoria cache. El tamaño de la cache debe estar relacionado con el de las rutinas o programas que se pretende acelerar. El software más moderno, de mayores dimensiones, lógicamente requiere tamaños de cache superiores a los mencionados anteriormente (por ejemplo: es común usar varios MB en la actualidad).

Hace muchos años que no se usan las memorias bipolares. En la actualidad las memorias son todas MOS tanto la principal como la cache. Pero dado que la primera es dinámica para lograr alta capacidad, usando un solo transistor y un capacitor, la segunda es estática, usa 6 transistores sin capacitor y por tanto es mucho más rápida, aunque más costosa por bit.

1.5.3 Configuración avanzada

A medida que el tiempo de acceso de las memorias disminuía, la configuración circuital de la figura 1.5.1 tropezó con un inconveniente físico: la línea de transmisión que constituye el bus del sistema.

Para reducir los efectos de la capacidad e inductancia de las conexiones, se diseñó un bus privado, sólo para la memoria principal y la cache, que además *duplica* el número de líneas de datos (véase la figura 1.5.3). En 1996 la memoria principal de una PC de oficina tenía un tiempo de acceso de 60 ns y la memoria cache uno de 15 ns.

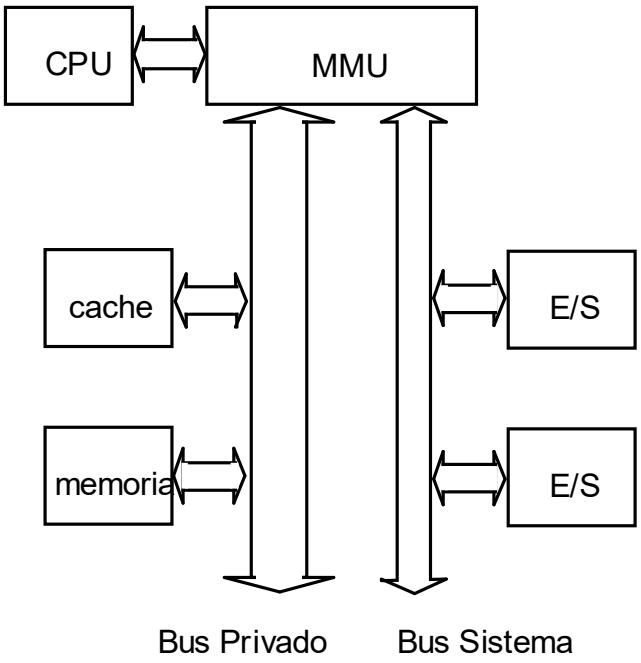


Fig. 1.5.3 - Bus privado

En los procesadores iAPX486 de (1989), Pentium (1992), Pentium Pro (1995) y posteriores, la unidad de manejo de memoria (MMU), la memoria cache y el bus privado se encuentran alojados en el chip del procesador, incrementando notablemente la performance.

Actualmente las caches se organizan en distintos niveles y además se distinguen según estén asociadas a datos o direcciones. Esto último facilita el acceso a los pipelines de búsqueda. Las más próximas al procesador se denominan L1 (Level 1: Nivel 1) y si existe otra a continuación, como en los procesadores actuales, se denomina L2 (Level 2: Nivel 2). Eventualmente puede existir un nivel L3 en los procesadores de múltiples núcleos.

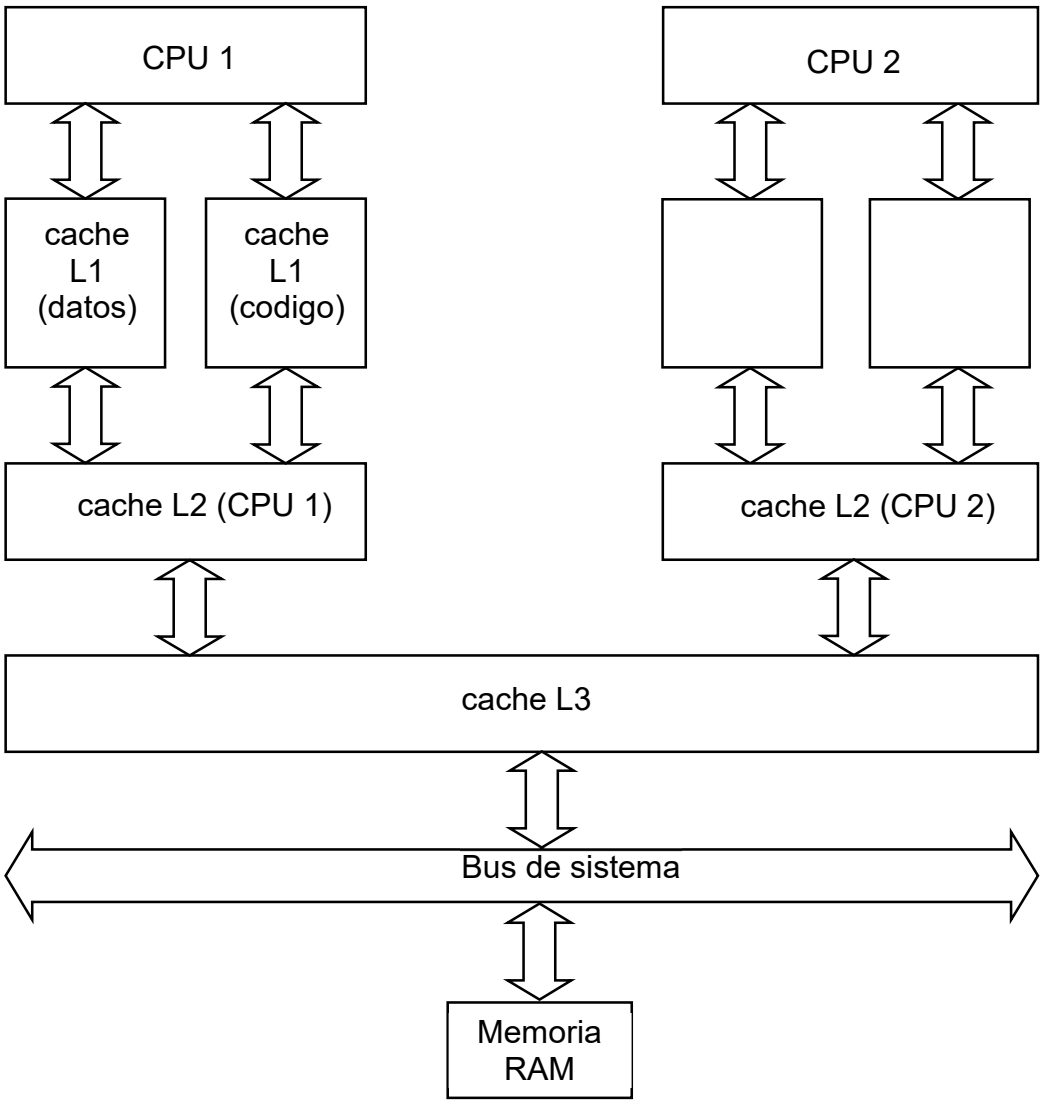


Fig. 1.5.4 – Niveles de cache

El Core I7 de Intel tiene 4 núcleos, cada uno de los cuales posee 32 kB de cache L1 para datos y 32 kB para código, 256 kB de cache L2 por núcleo y 8000 kB de cache L3 compartida por los 4 núcleos. Esto suma: 8x32 + 4x256 + 8000 = 9280 kB. Esta cantidad de cache utiliza más de 450 millones de transistores, de los 731 millones que posee este procesador.

El Phenom de AMD tiene también 4 núcleos, cada uno de los cuales posee 64 kB de cache L1 para datos y 64 kB para código, 512 kB de cache L2 por núcleo y 2000 kB de cache L3 compartida por los 4 núcleos. Esto suma: 8x64 + 4x512 + 2000 = 4560 kB. Esta cantidad de cache utiliza 224 millones de transistores, de los 450 millones que posee este procesador.

Los procesadores actuales de Intel pueden tener hasta 18 núcleos y 40 MB de caché L3 y los de AMD pueden tener hasta 64 núcleos y 256 MB de caché L3.

En el apéndice E se pueden ver las características de los principales procesadores Intel, AMD e IBM.

1.6 SISTEMAS MULTIUSUARIO

1.6.1 Sistema multitarea

Es un sistema que da la sensación de ejecutar varias tareas (programas) al mismo tiempo. Requiere una implementación de hardware: sistema de interrupciones y una de software: sistema operativo multitarea.

Las tareas se ejecutan basándose en un sistema de prioridades, donde la tarea más prioritaria se ejecuta hasta que requiera un recurso no disponible o hasta que finalice. En ese momento se transfiere el control a la que sigue y así sucesivamente hasta agotar la lista. Como todos los programas requieren de algún acceso a disco, impresora, pantalla, etc. y estos periféricos son más lentos que la CPU, queda tiempo de máquina durante el cual avanza la ejecución de las tareas menos prioritarias. Este esquema de planificación (scheduling) se conoce como "FCFS" (First Come, First Served).

En los procesadores actuales, para asegurar que se respeten las prioridades con una asignación de tiempo de CPU proporcional a la prioridad establecida, el RTC (reloj de tiempo real) genera una interrupción no enmascarable (NMI) cada 1 ms lo que obliga al sistema operativo a evaluar la lista de programas en espera de ejecución y los tiempos que les va asignando. En la figura 1.6.1 se ve la asignación de tiempos para 3 programas (#A #B #C) cuyas prioridades fueron fijadas en 5, 2 y 1 respectivamente. En Windows las prioridades tienen un rango de 1 a 255, siendo 255 la más alta.

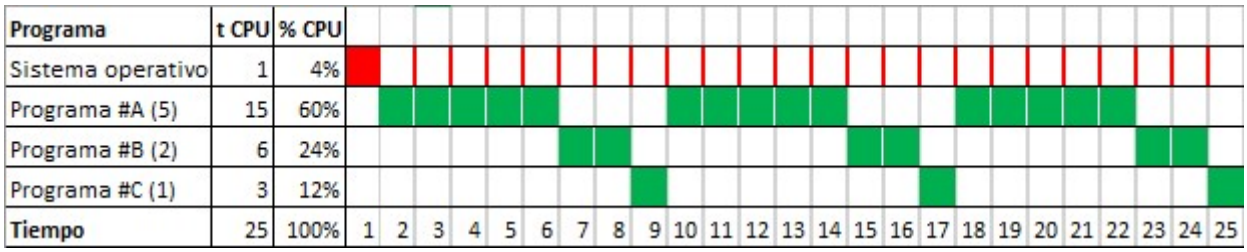


Fig. 1.6.1 – Sistema multitarea – diagrama de tiempos

Un algoritmo de planificación más avanzado es el denominado "round-robin", donde se agrega al esquema anterior la posibilidad de cambiar de tarea luego de agotado un tiempo preestablecido, denominado "time quantum" (entre 10 y 100 ms). En este caso, se programa un temporizador, que al cumplir el tiempo fijado genera una interrupción de HW, la que restituye el control al sistema operativo. La tarea en curso es enviada al final de la lista de pendientes de ejecución, pasándose el control a la primera de la lista. Si la tarea se ejecuta dentro del tiempo fijado, se sigue usando FCFS.

1.6.2 Sistema multiusuario

Es un sistema en el cual los usuarios tienen la sensación de disponer la computadora para ellos, pero en realidad el SO administra los recursos y los usuarios compiten por ellos. En estos sistemas, se asignan a los usuarios privilegios para tener acceso a los recursos.

El administrador del sistema es un usuario con los máximos privilegios. Él tiene la función de asignar a los usuarios los recursos del sistema (cuentas, espacio en disco, tiempo de CPU, dispositivos de impresión, mensajes, nivel de seguridad, recursos de software, etc.).

Para poder ser multiusuario, el sistema debe ser multitarea. Además, requiere una implementación en el hardware: los modos protegido y usuario/supervisor y en el software: el sistema operativo multiusuario.

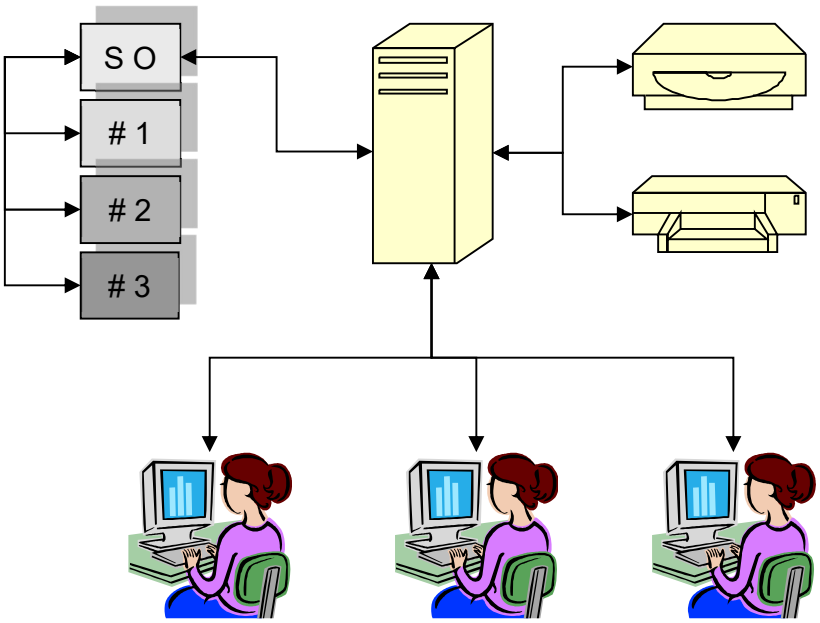


Fig. 1.6.2 – Sistema multiusuario

El sistema debe tomar los recaudos para que un usuario no afecte (intencionalmente o no) a otros usuarios o a la integridad del sistema. Por ejemplo, si un usuario escribe un programa en assembly que incluya la instrucción HLT (Halt Processor), hasta el momento de la ejecución nadie se daría cuenta de lo que pretende hacer. Si esta instrucción se ejecutase, se detendría la CPU y todos los usuarios se verían perjudicados.

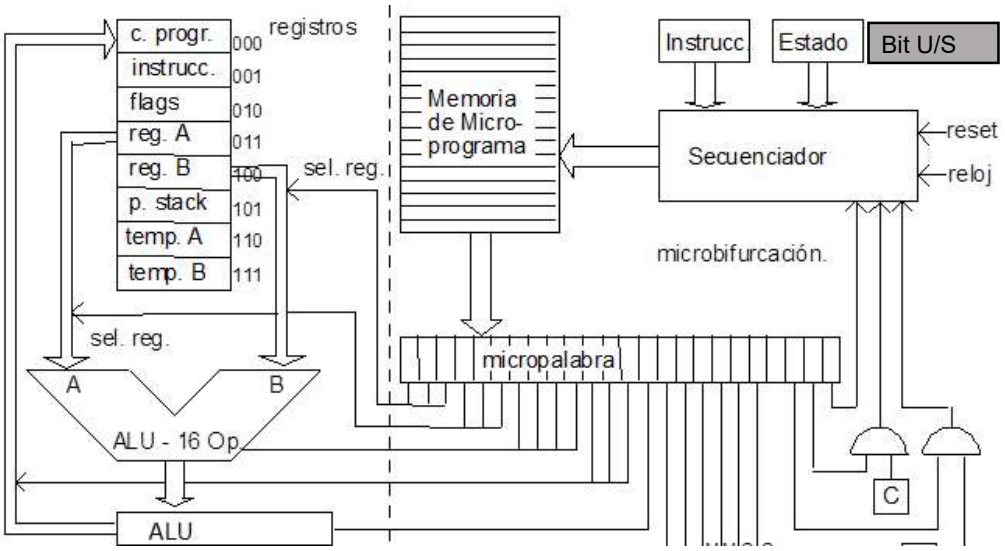
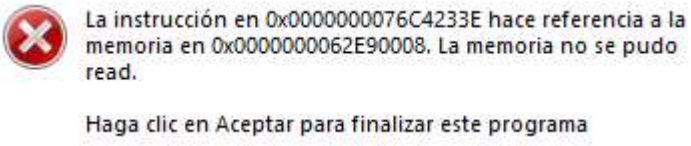


Fig. 1.6.3 – Modo U/S en la arquitectura microprogramada

En un procesador de estas características, la CPU toma las instrucciones de memoria de la manera habitual, las descodifica y en caso de ser HLT, se fija en el bit de usuario/supervisor del estado del procesador (PSW). Si está en supervisor, ejecuta HLT de la manera normal, deteniendo la CPU. Si está en modo usuario, genera una excepción de instrucción inválida, conmuta la CPU a modo supervisor y da la posibilidad al SO de suspender la tarea en cuestión sin que los demás usuarios se percaten de lo que ha ocurrido. Otras instrucciones que generan esta excepción son: RESET y modificación del PSW.

Para prevenir la situación anterior, el hardware provee dos modos de trabajo, usuario y supervisor y se asigna un bit en la palabra de estado del procesador PSW para indicar en qué modo se encuentra la CPU en todo momento. Los programas de usuario se ejecutarán en este modo y el SO en el modo supervisor.

Por último, en los sistemas multiusuario existen registros que determinan los derechos de acceso a las zonas de memoria. Esta implementación de hardware impide a un usuario escribir en la memoria de otro o en la del SO, generando una excepción de dirección inválida y permitiendo al SO suspender la tarea transgresora, como muestra esta alerta de Windows.



Los archivos en medios magnéticos están protegidos por software, pero un usuario no puede cambiar los derechos de acceso de los archivos de otro y por ende si el otro determinó que no puedan ser leídos por terceros, no lo serán.

De la misma forma, el SO maneja el servicio de impresión, para evitar que se mezclen las páginas que desean imprimir los distintos usuarios.

A continuación, se puede ver una tabla que resume lo visto precedentemente.

Recurso compartido	Implementación de HW	Implementación de SW	8086	80386
	¿Se requiere un HW adicional para administrar los permisos de acceso al recurso? ¿Cuál?	¿Cómo implementa el Sistema operativo los permisos de acceso al recurso?	¿se puede administrar completamente el recurso con este procesador?	¿se puede administrar completamente el recurso con este procesador?
Disco rígido	no, es suficiente con el sist. de interrupciones	con el administrador de archivos	si	si
Impresora	no, es suficiente con el sist. de interrupciones	con el administrador de impresión	si	si
Red	no, es suficiente con el sist. de interrupciones	con el administrador de red	si	si
Memoria	sí, el modo protegido	Sistema operativo multiusuario	no (no tiene MP)	si
Procesador	sí, el modo usuario/supervisor	Sistema operativo multiusuario	no (no tiene U/S)	si

Fig. 1.6.4 – Resumen de requisitos de HW y SW

1.7 MODO PROTEGIDO

1.7.1 Microprocesador provisional

Cuando Intel presentó por primera vez el 8086, este fue conocido internamente como "interim microprocessor" (la CPU que usarían los usuarios de 8080, 8085 y Z80 para actualizarse hasta tanto el iAPX 432, de 32 bits estuviese listo para la venta. Así, la primera consideración de diseño del 8086 fue que los programas escritos para los viejos microprocesadores de 8 bits fueran fáciles de transportar al 8086. Los diseñadores dijeron que tanto los juegos de registros como el conjunto de instrucciones permitían traducir *mecánicamente* las aplicaciones de 8080 en aplicaciones para el nuevo microprocesador.

Los pares de registros, que podían ser manipulados como una unidad en la arquitectura anterior, ahora eran de 16 bits de ancho. Ya que el 8086 tenía 20 bits de direcciones, las aplicaciones transportadas deberían haberse tenido que rescribir extensivamente si los cálculos de direcciones se hubiesen hecho en registros mayores. Esta es una de las razones de por qué Intel eligió dividir las direcciones en dos partes: un *registro de segmento* de 16 bits y un *desplazamiento de dirección efectivo* (o dirección lógica).

Como se vio en el ejemplo del capítulo 1.4, el 8086 tiene cuatro registros de segmento, los cuales fueron usados para mapear las direcciones de 16 bits utilizadas por la pila, datos y código del programa en cuatro regiones potencialmente disjuntas en el espacio de direcciones de 1 Mbyte.

El registro de segmento es desplazado cuatro bits a la izquierda y sumado a la dirección lógica para completar la dirección física de 20 bits.

Aunque esta arquitectura no hace fácil manipular estructuras de datos mayores que 64K bytes, poca gente tuvo esa cantidad de RAM en esa época.

Fue fácil para muchos programas (y virtualmente todas las aplicaciones transportadas desde CP/M) fijar los registros de segmento por única vez al principio del programa y olvidarse de ellos. Podría decirse que el contenido de los registros de segmento provee valores por defecto para los bits superiores de cada dirección. Desde entonces, el programa trató principalmente con desplazamientos de dirección efectivos de 16 bits, la mitad de tamaño de las direcciones manejadas por el 68000. El conjunto de instrucciones era también más compacto y no requería que las instrucciones o datos estuviesen alineados en el límite de palabras (de 16 bits). Como resultado, el código y los datos inmediatos demandaron cerca del 30% menos espacio. En aquellos días, la RAM era costosa, una buena razón para IBM para elegir la arquitectura Intel sobre la de MOTOROLA para la IBM PC.

1.7.2 El 80286: Primer Intento de Modo Protegido

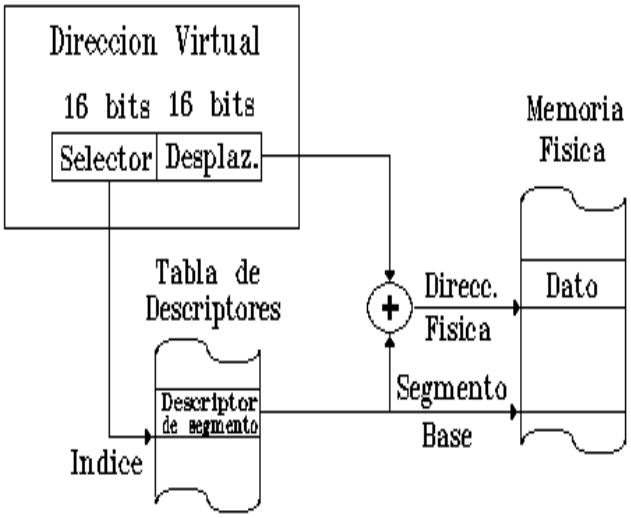
Desafortunadamente, el iAPX 432, un microprocesador de 32 bits en dos chips diseñado para multiprocesamiento, multitarea, programación orientada a objetos y tolerancia a fallas de hardware y software, no fue un éxito en el mercado. Esto puede haber sido debido a que fue demasiado adelantado para su tiempo, o debido a que fue concebido para correr ADA (el cual no logró la penetración en el mercado que Intel pensó que tendría). O tal vez, simplemente, los chips fueron demasiado costosos.

Pero el IBM PC y el 8088 realmente tuvieron éxito y los usuarios clamaron por más memoria y por más potencia de procesamiento para correr su software existente. Con el 8086 Intel les dio todo eso y más. Tal como el resto de este artículo mostrará, el 80286 incluyó muchas de las características implementadas originalmente para el iAPX 432.

Dos características son esenciales para cualquier sistema que provea protección razonable contra programas errantes y usuarios maliciosos: *aislación de tareas* y *protección de los recursos del sistema*. El sistema necesita prevenir que las tareas corrompan recíprocamente sus memorias o acaparen el sistema. Los recursos del sistema (p/ej. hardware y sistema de archivos) requieren permanecer bajo el control exclusivo del software del sistema operativo. Puesto que la mayoría de los recursos son accesibles tanto a través de direcciones de memoria como de E/S, el sistema debe mantener control sobre la manera en que una tarea accede a ellos.

1.7.3 Direccionamiento Virtual

El 80286, luego de una condición de Reset, arranca en modo real. Una rutina de inicialización, la cual se ejecuta en modo real, define *tablas de descriptores*, que gobiernan como es direccionada la memoria y luego conmutan la CPU al modo protegido fijando un bit en un registro de control. Intel hizo esta transición casi irreversible en el 80286. El 80386, sin embargo, puede ser retrotraído al modo real reponiendo el mismo bit. Esto permite a la sesión en modo DOS del OS2 conmutar modos literalmente sin detener el microprocesador.



DIRECCIONAMIENTO VIRTUAL EN MODO PROTEGIDO

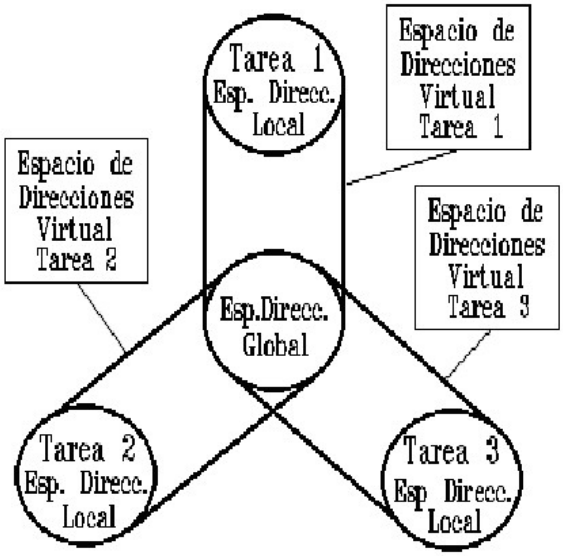
Fig. 1.7.1

En modo protegido, no hay más una simple relación aritmética entre un selector de segmento, el desplazamiento y la dirección física a la que ellos refieren. En lugar de eso, el microprocesador interpone una operación de mapeo especial entre una *dirección virtual* (el selector de segmento del modo protegido y el desplazamiento, tomados juntos) y la dirección física, como se muestra en la figura 1.7.1.

En modo protegido, un selector de segmento no es un agregado de una dirección; en vez de ello es una "receta mágica". El selector refiere a un *descriptor de segmento*, un registro en memoria conteniendo información acerca del segmento, incluyendo la *dirección base del segmento*, en la que el segmento comienza. El desplazamiento de dirección efectivo es sumado a la dirección base para producir una dirección física. El puntero de *selección* para todos los chips 80x86 es de 16 bits de ancho. El puntero de *desplazamiento* del 80286 es de 16 bits de ancho. Sin embargo, el puntero de desplazamiento para el 80386 y 80486 es de 32 bits de ancho.

1.7.4 Espacios de Direccionamiento Local y Global

Cada tarea que corre en modo protegido puede acceder segmentos dentro de un espacio de direccionamiento virtual que consiste en dos espacios menores: su *espacio de direcciones local* y su *espacio de direcciones global* (ver la figura 1.7.2). Los objetos de código y datos necesitados por toda tarea en el sistema, son puestos en el espacio de direcciones global, mientras que los objetos usados por una sola tarea van en el espacio de direcciones local. Ya que una tarea no puede acceder a memoria para la cual no tiene un selector, los espacios de direcciones locales están garantizados de ser privados a menos que un segmento sea mapeado intencionalmente en dos o más de ellos.



ESPACIO DE DIRECCIONES VIRTUAL Y AISLACION DE TAREAS

Fig. 1.7.2

La figura 1.7.3 muestra el formato de un selector de modo protegido y la forma en que este es usado para encontrar el descriptor de un segmento en memoria. Los 13 bits superiores del selector dan la ubicación del descriptor en un arreglo de descriptores llamado tabla de descriptores. Una tabla de descriptores es de 8 bytes de longitud, de forma que enmascarando los tres bits inferiores del selector convenientemente se forma el desplazamiento en la tabla del byte del descriptor.

Los tres bits inferiores del selector son bits de control. El bit 2, llamado el *indicador de tabla* (TI), toma una de las dos tablas de descriptores posibles: la *tabla de descriptores global*, la cual mantiene descriptores para segmentos en el espacio de direcciones global, o la *tabla de descriptores local* de la tarea, que hace lo mismo para el espacio de direcciones local.

Los bits restantes, 1 y 0, son el campo de *nivel de privilegio requerido (RPL).* Estos bits describen el nivel de privilegio que una tarea desea que se conceda cuando se accede al segmento.

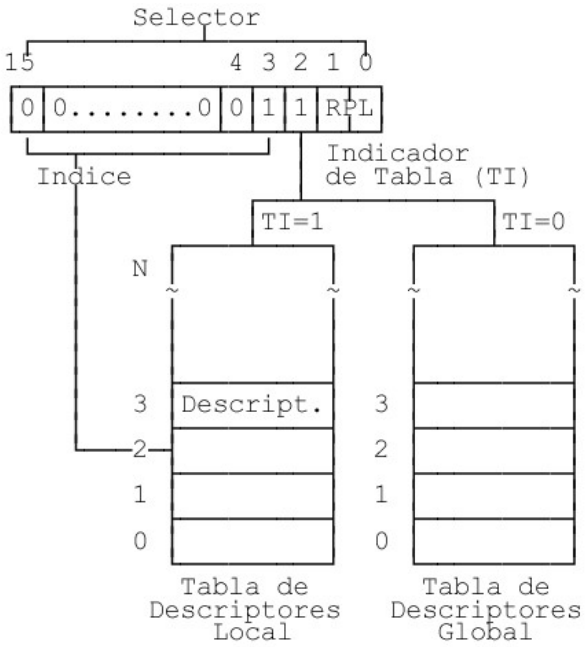


Fig. 1.7.3

1.7.5 Manejo de Memoria Sin Preocupaciones

Los selectores de segmento y las tablas de descriptores toman gran parte de la preocupación del manejo de memoria. Debido a que un selector de segmento es, en esencia, un "handle" que aparece como doble referenciado durante cada acceso a memoria, los segmentos pueden ser movidos sobre la memoria física en forma invisible. Y ya que un descriptor de segmento es automáticamente capturado dentro del microprocesador cuando su selector es puesto en un registro de segmento, no se necesitan dos registros de direcciones para acceder al segmento potencialmente movible. La carga de un selector de 16 bits lo hace todo.

Un segmento en modo protegido no sólo puede ser movido de un lugar a otro dentro de la RAM física, también puede ser permutado enteramente fuera de la RAM. El mecanismo de selector/descriptor de segmento permite al 80286 y sus parientes proveer memoria virtual sin una unidad de manejo de memoria separada. Cada descriptor de segmento contiene un bit que indica si el segmento está o no presente físicamente en RAM, y el microprocesador genera una excepción si una aplicación intenta acceder a un segmento que ha sido purgado o permutado hacia afuera.

1.7.6 Protección: el Modelo de la Cebolla

La jerarquía de privilegios y protección en los microprocesadores Intel sigue el modelo del *anillo*, o *cebolla*, mostrado en la figura 1.7.4. Hay cuatro niveles de privilegio posibles, numerados de 0 a 3, donde el nivel 0 es el más privilegiado.

Toda tarea en el sistema tiene un *nivel de privilegio actual (CPL)*, el cual es usado para determinar si puede hacer E/S, ejecutar instrucciones privilegiadas (aquellas que hacen cosas tales como parar la CPU), y/o acceder segmentos. Cada selector de segmento tiene también un campo de RPL, el cual es cambiado por una tarea si ella desea tener menor privilegio del que su CPL normalmente permitiría cuando accede al segmento. El *nivel de privilegio efectivo (EPL)* es menos privilegiado que el RPL y el CPL, y el determina si la tarea puede acceder a un segmento con un selector dado.

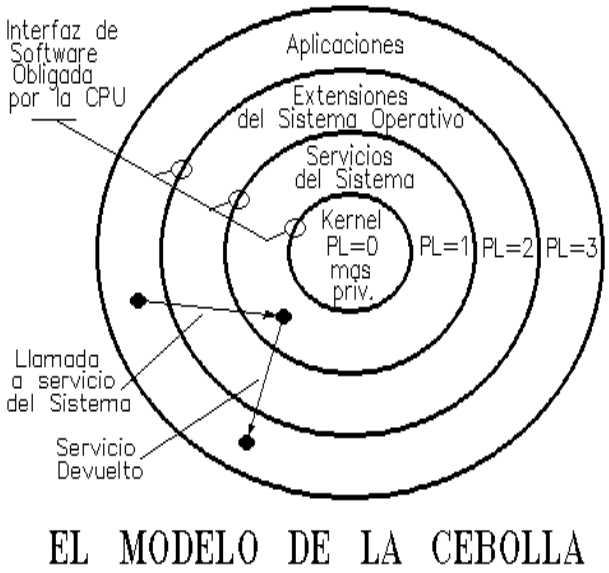


Fig. 1.7.4

¿Por qué querría una tarea darse a sí misma menos privilegio del que podría tener en otra ocasión al acceder a un segmento?. La respuesta se logrará con algo llamado: el problema del caballo de Troya. Suponga que un sistema operativo de modo protegido tiene una función llamada *DOSREAD*, la cual lee información de un archivo abierto (el OS2 tiene una función tal como ésta). La definición (encabezamiento) de esta función en C habrá de verse así:

```
unsigned int pascal far DOSREAD
(unsigned short FileHandle,
        void far*PtrBuffer,
    unsigned int BufferLength,
  unsigned int far*PtrBytesRead);
```

Usted no tiene que ser hábil en C para el propósito de este ejemplo; lo importante es que, entre los argumentos, hay un puntero llamado *PtrBuffer* (declarado como *void far*PtrBuffer*). La palabra *void* indica que el puntero es indefinido (ej.: puede apuntar a un objeto de cualquier clase), y la palabra *far* indica que el puntero consiste tanto de selector de segmento como de desplazamiento. *PtrBuffer* da la dirección de la locación que va a recibir los datos de la operación de lectura.

Ahora, suponga que una aplicación llamó a la función PtrBuffer apuntando a un segmento para el que a la aplicación se le ha permitido acceder (porque ésta no ha tenido el privilegio suficiente). El sistema operativo, que presumiblemente correría en un nivel de privilegio mayor, tendría acceso a ese segmento, así la aplicación causaría, en teoría, que el sistema operativo lea arbitrariamente datos en ese lugar de la memoria.

Para prevenir que esto ocurra, el sistema operativo ajusta el campo RPL dentro de la parte del selector de *PrtBuffer* y así él no tiene más privilegio que la aplicación cuando accede al segmento. Entonces, si el puntero se torna un caballo de Troya, la violación de privilegio será capturada por los mecanismos de protección habituales.

Toda la información acerca de que puede ser hecho con un segmento, y por quien, está contenida en el descriptor de segmento. Cada descriptor de segmento tiene un byte *de derechos*

de acceso, el cual a su vez contiene un campo de dos bits: el *descriptor de nivel de privilegio*. El EPL de la tarea debe darle igual o mayor privilegio que el DPL para que un acceso sea permitido. El byte de derechos de acceso también contiene otra información acerca de que puede ser hecho con el segmento, si puede ser leído o escrito, por ejemplo, y si este contiene código, datos, o una pila de una tarea.

Bibliografía:
L. Brett Glass – BYTE (Diciembre 1989) – Traducción y adaptación: FFP
Consulte también el texto de Tanenbaum "Organización de Computadoras".

1.7.7 Protección: el 80386

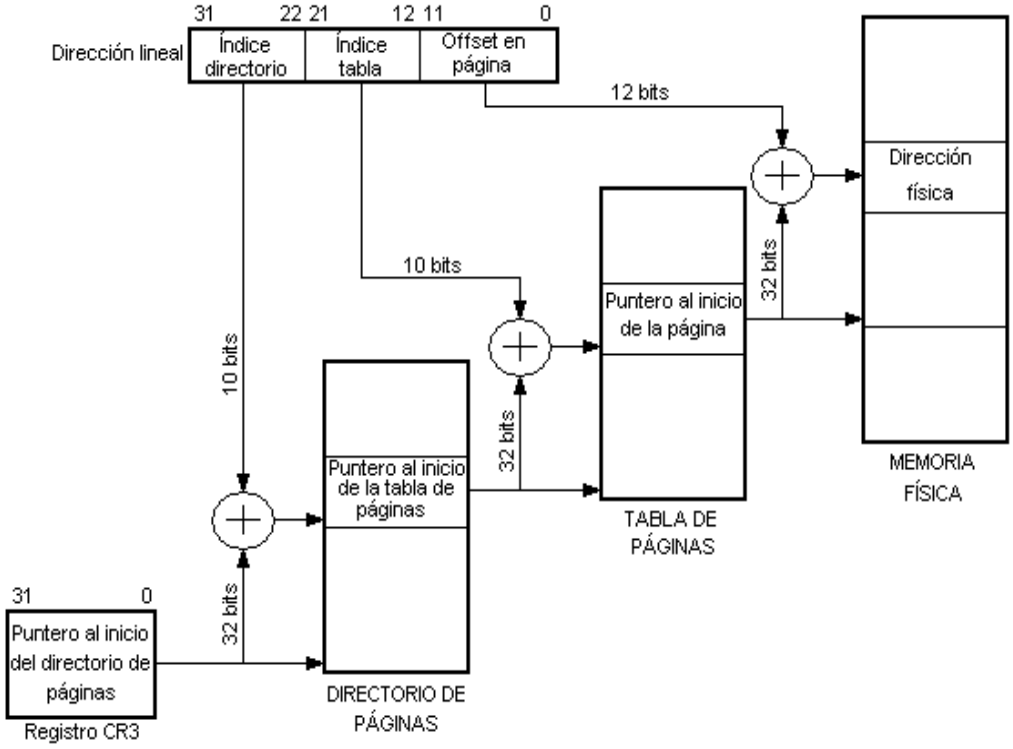


Fig. 1.7.5 – Paginación en el 80386

En el procesador 80386 la protección de la memoria basada en la paginación se implementa de acuerdo con el esquema de la figura 1.7.5.

El registro CR3 contiene la dirección física de donde comienza el directorio de páginas. Los 12 bits menos significativos del registro están en cero para que el directorio de páginas esté en una misma página.

El tamaño del directorio de páginas es de 4 kB y permite hasta 1024 entradas. Estas entradas se seleccionan mediante los bits 31-22 de la dirección lineal.

Cada entrada del directorio contiene la dirección de la tabla de páginas y varios bits de atributos de la página, como se muestra a continuación:

31 – 12	***11 - 9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***

Dirección tabla de páginas (31-12)	**Libre**	**0**	**0**	**D**	**A**	**0**	**0**	**U/S**	**R/W**	**P**

El tamaño de las tablas de páginas es de 4 kB y permite hasta 1024 entradas. Estas entradas se seleccionan mediante los bits 21-12 de la dirección lineal. Cada entrada de la tabla contiene la dirección inicial de la página y varios bits de atributos, como se muestra a continuación:

31 – 12	***11 - 9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Dirección física de página (31-12)	**Libre**	**0**	**0**	**D**	**A**	**0**	**0**	**U/S**	**R/W**	**P**

Los 20 bits más significativos de la dirección de la página se suman con los 12 bits menos significativos de la dirección lineal para formar la dirección física.

El bit 0 (Presente), cuando está en 1 indica que la página está en memoria y por lo tanto la entrada se puede utilizar para traducir de dirección lineal a física. Cuando P = 0 los otros 31 bits quedan libres a disposición del programador. Por ejemplo, si se usa memoria virtual, estos bits podrían utilizarse para indicar dónde se encuentra la página en el disco.

El bit 5 (Accedido), es puesto a uno por el 80386 en ambos tipos de entradas cuando ocurre un acceso de lectura o escritura en una dirección que esté dentro de una página cubierta por estas entradas.

El bit 6 (Dirty), es puesto a uno por el microprocesador cuando ocurre un acceso de escritura.

Los tres bits marcados como libres pueden ser utilizados por el programador.

Los bits 1 (Lectura/Escritura) y 2 (Usuario/Supervisor) se utilizan para proteger páginas individuales.

Para la paginación existen dos niveles de protección: usuario que corresponde al nivel de privilegio 3 y supervisor, que corresponde a los otros niveles: 0, 1 y 2.

Los bits R/W y U/S se utilizan para proveer protección de Lectura/Escritura y Usuario/Supervisor para páginas individuales o para todas las páginas cubiertas por una tabla de páginas. Esto se logra tomando los bits R/W y U/S más restrictivos entre el directorio de páginas y la tabla de páginas que corresponda a la página en cuestión.

Bibliografía:
Darío Alpern – Adaptación: FFP

1.8 MEMORIA Y MÁQUINA VIRTUALES

Los MC68010/MC68012 introdujeron el concepto de memoria y máquina virtual en la arquitectura de la familia 68000 de Motorola. En la familia 80x86 de Intel los primeros procesadores en incorporar estas características fueron el 80286 y el 80386.

En la mayoría de los sistemas de micro cómputo, sólo una fracción del espacio de direcciones corresponderán realmente a memoria física. Con las técnicas de **memoria virtual**, el sistema puede ser hecho de manera que para el usuario aparezca como teniendo más memoria física accesible. Estas técnicas han sido usadas durante muchos años en grandes computadoras, luego en minicomputadoras y desde la década del 1980 está disponible en los microprocesadores.

En un sistema de memoria virtual, un programa de usuario puede ser escrito pensando que hay una gran cantidad de memoria disponible para él, cuando sólo una pequeña cantidad de memoria está presente físicamente en el sistema. En una forma similar, un sistema puede ser diseñado de tal manera que permita a los usuarios acceder a otros tipos de dispositivos que no están físicamente presentes en el sistema, tales como unidades de cinta, unidades de disco, impresoras o CRTs.

Con una apropiada emulación de software, un sistema físico puede ser hecho de manera que aparezca al programa de usuario como algún otro sistema de computadora y el programa puede tener acceso a la totalidad de los recursos del sistema emulado. Tal sistema emulado es llamado una **máquina virtual**.

1.8.1 Memoria virtual

El mecanismo básico para soportar memoria virtual en las computadoras es proveer una cantidad limitada de memoria física de alta velocidad que puede ser accedida directamente por el procesador, mientras se mantiene una imagen de memoria "virtual" mucho mayor en dispositivos de almacenamiento secundarios tales como unidades de disco de alta capacidad. Cuando el procesador intenta acceder a una locación en el mapa de memoria virtual que actualmente no se encuentra en la memoria física (referido como falla de página), el acceso a esa locación es suspendido temporariamente mientras son buscados los datos necesarios desde el medio de almacenamiento secundario y puestos en memoria física. El acceso que había sido suspendido es entonces completado.

Los MC68010/MC68012 proveen soporte de hardware para memoria virtual con la capacidad de suspender la ejecución de una instrucción cuando se señaliza un error de bus y luego completar la instrucción, luego que la memoria física haya sido actualizada como era necesario.

Los MC68010/MC68012 usan continuación de instrucción en vez de reiteración de instrucción para soportar memoria virtual. Con reiteración de instrucción, el procesador debe recordar el estado exacto del sistema antes de que comience cada instrucción a los fines de restaurar ese estado en el caso de que ocurra una falla de página durante su ejecución. Entonces, después que la falla de página haya sido reparada, la instrucción que ha causado la falla se ejecutará completa. Con continuación de instrucción, cuando ocurre una falla de página, el procesador almacena su estado interno y luego de que la falla de página es reparada, restablece ese estado interno y continúa la ejecución de la instrucción. Para implementar la continuación de instrucción, los MC68010/MC68012 almacenan su estado interno en la pila del modo supervisor cuando un ciclo de bus termina con una señal de error de bus. Este carga luego el contador de

programa desde el vector número dos de la tabla de interrupciones (desplazamiento $008) y continúa la ejecución del programa a partir de esa nueva dirección. Cuando la rutina de manejo de excepciones correspondiente al error de bus ha completado su ejecución, se ejecuta una instrucción RTE, la cual carga el MC68010/MC68012 con el estado interno almacenado en la pila, ejecuta nuevamente el ciclo de bus que falló y continúa la instrucción suspendida. La continuación de instrucción tiene la ventaja adicional de permitir soporte de hardware para dispositivos virtuales de E/S. Como los registros virtuales pueden ser simulados en el mapa de memoria, un acceso a tales registros causará una falla de bus y la función de ese registro podrá simularse por software.

Ejemplo de memoria virtual con un procesador genérico

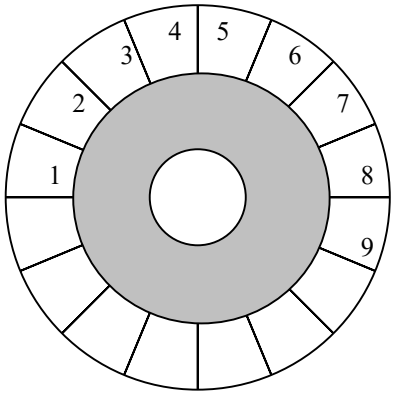
Se pretende cargar un programa que ocupa 9 bloques desde la unidad de disco. La memoria RAM disponible (L) es de 6 bloques. Dos bloques están ocupados por el sistema operativo (SO).

0	0	N
1	0	N
2	0	N
3	0	N
4	0	N
5	0	N
6	0	N
7	0	N
8	0	N
9	0	N

Tabla de Mem. Virtual

00	SO
10	SO
20	L
30	L
40	L
50	L
60	L
70	L

Memoria RAM



Unidad de disco

Fig. 1.8.1 – Memoria virtual - problema

Si el sistema no posee HW para soportar memoria virtual, el SO informará “memoria insuficiente” y el programa no podrá ejecutarse. En cambio, si el HW está disponible, el procesador contará con registros para “cargar” el programa completo como se ve en la figura 1.8.1. Primero el SO lee el bloque 1 del disco, donde está la longitud total del archivo (cantidad de bloques). Luego verifica cuantos bloques libres de memoria hay y copia del disco los primeros 6 bloques hasta completar la memoria. Finalmente, completa la tabla de la memoria virtual indicando los bloques presentes (P) y los que no se pudieron traer (N).

La tabla y la memoria, luego de la carga inicial, quedan como muestra la figura 1.8.2. Se consideró una tabla con 10 posiciones, donde se cargó la dirección de cada bloque, ya sea en memoria, si está presente o en disco, si no lo está.

Ahora el SO transfiere el control al programa, en la dirección de memoria 20. Para acceder a esta dirección el procesador determina la dirección física basado en la tabla de direcciones TMV, y si está presente comienza a ejecutar normalmente. Si en algún momento se pretende un acceso a una dirección no presente, por ejemplo, el bloque 7 del disco, se genera una excepción (interrupción interna) cuya rutina de atención deberá traer del disco el bloque 7, actualizar la tabla TMV y retomar el programa, ahora con el bloque 7 en RAM.

Hay distintas estrategias para determinar que bloque de RAM se libera para hacer lugar al nuevo bloque a traer de disco. Una de ellas, FIFO, consiste liberar el bloque más antiguo en RAM.

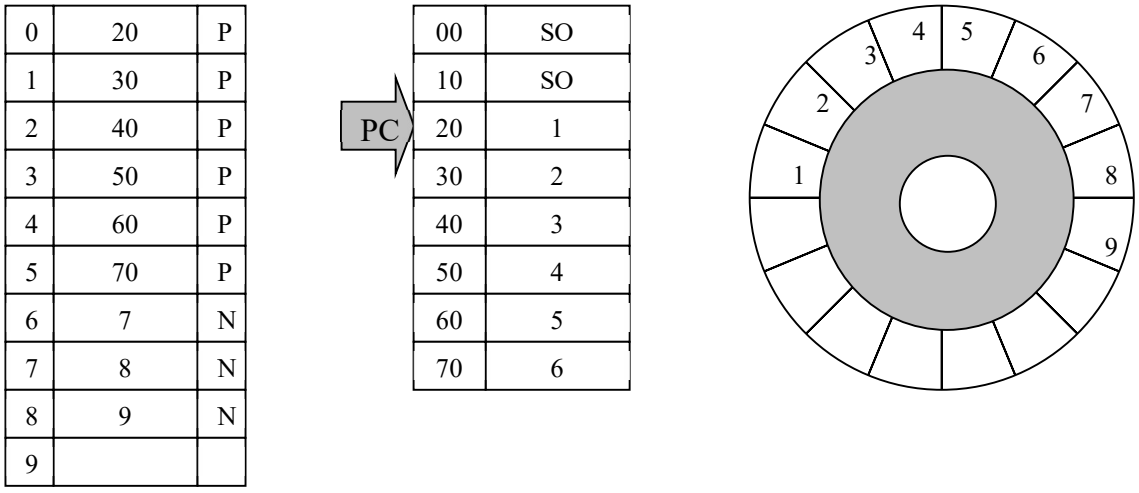


Fig. 1.8.2 – Memoria virtual - carga inicial

1.8.2 Máquina virtual

Un uso típico de un sistema de máquina virtual es en el desarrollo de software, como puede ser un sistema operativo, para otra máquina con hardware aún en desarrollo y por ende no accesible para el programador. En tal sistema, el sistema operativo director emula el hardware del nuevo sistema operativo sea ejecutado y depurado tal como si hubiese estado corriendo en el nuevo hardware. Como el nuevo sistema operativo está controlado por el sistema operativo director, el nuevo debe ejecutarse con un nivel de privilegio menor que el sistema operativo director, de forma tal que cualquier intento del nuevo sistema operativo de usar recursos virtuales que no estén físicamente presentes, y que deberían ser emulados, será capturado por el sistema operativo director y manejado por software.

En los MC68010/MC68012 puede ser completamente soportada una nueva máquina virtual, corriendo el nuevo sistema operativo en el modo usuario y el sistema operativo director en el modo supervisor, de forma que cualquier intento de acceder a los recursos del supervisor o ejecutar instrucciones privilegiadas por el nuevo sistema operativo será capturado (trap) por el sistema operativo director.

A los efectos de soportar completamente una máquina virtual el MC68010/MC68012 debe proteger los recursos del supervisor del acceso de por parte de programas de usuario. El único recurso de supervisor que no está completamente protegido en el MC68000 es el byte del sistema del registro de estado. En el MC68000 y el MC68008, la instrucción MOVE desde el SR permite a los programas de usuario verificar el estado del bit S (además de los bits T y la máscara de interrupciones) y así determinar que ellos están corriendo en el modo usuario.

Para un completo soporte de máquina virtual, un sistema operativo nuevo no debe estar enterado del hecho que correrá en el modo usuario y por lo tanto no debería permitirse que lea el bit S. Por esta razón se ha agregado la instrucción MOVE desde el SR para permitir al programa de usuario un libre acceso a los códigos de condición. Si la instrucción MOVE desde el SR

hubiese sido privilegiada, cuando el sistema operativo nuevo intente acceder al bit S, será capturado por el sistema operativo director y la imagen del SR que éste le pasará contendrá el bit S fijado (verdadero).

Bibliografía:
MC68010/MC68012 16/32 Virtual Memory Microprocessor. (traducción: FFP)

1.9 SOFTWARE PARA EJECUTAR MÁQUINAS VIRTUALES

INTRODUCCIÓN:

Existen programas que, ejecutándose en un procesador que soporte modo protegido y con el sistema operativo apropiado, pueden ejecutar múltiples máquinas virtuales en una única máquina real. Tal es el caso de VMware, Microsoft Virtual PC y VirtualBox.

Además de su interés como medio de desarrollo, experimentación y estudio, actualmente la virtualización permite también la racionalización del hardware de servidores, instalando en una sola máquina anfitrión lo que antes se hacía con varias, cada una de ellas con su diferente HW y sistema operativo. Finalmente, esto ayuda también a la racionalización energética por tener menos nodos físicos activos y al menor costo de mantenimiento por la estandarización del HW.

1.9.1 VirtualBox - Arquitectura

La virtualización es, por naturaleza, extraordinariamente compleja, especialmente en hardware basado en x86. La comprensión del código fuente de VirtualBox por lo tanto requiere al menos para algunos componentes, un gran entendimiento acerca de los detalles de la arquitectura x86 como así también un gran conocimiento acerca de la implementación de la plataformas anfitrión e invitado involucradas.

Los procesos VirtualBox:

Cuando usted inicia la interfaz gráfica VirtualBox, al menos un proceso adicional se inicia (VBoxSVC: el proceso de “servicio” VirtualBox). Una vez que usted ha iniciado la máquina virtual (VM) desde de la interfaz gráfica, usted tiene dos ventanas (y la ventana principal y de la VM), pero habrá tres procesos corriendo. Observando su sistema desde el administrador de tareas (en Windows) o el monitor de sistema (en Linux) usted verá esto:

1. VirtualBox, la interfaz gráfica para la pantalla principal.
2. Otro proceso VirtualBox que fue iniciado con el parámetro –startvm, lo cual significa que su interfaz gráfica actuará como un entorno para la máquina virtual.
3. VBoxSVC, el servicio mencionado previamente, el cual está corriendo en segundo plano para seguir la pista de todos los procesos involucrados. Esto fue automáticamente iniciado por el primer proceso de interfaz gráfico.

Nombre de imagen	Nombre de us...	ID. DE ...	CPU	Uso de ...
VBoxSVC.exe	Fernando	0	00	34.148 KB
VirtualBox.exe	Fernando	0	00	51.228 KB
VirtualBox.exe	Fernando	0	64	100.312...

Fig. 1.9.1 – Procesos de VirtualBox

Para el sistema operativo anfitrión (SO), la VM que corre "dentro" de la segunda ventana se ve como un programa ordinario. VirtualBox tiene muy buen comportamiento en ese aspecto: él toma el control sobre una gran parte de su computadora, ejecutando un sistema operativo completo con su propio conjunto de procesos invitados, drivers, y dispositivos adentro de su proceso de máquina virtual, pero el sistema operativo anfitrión no tiene demasiada noticia acerca de esto. Lo que sea que la máquina virtual haga, es sólo un proceso más en su sistema operativo anfitrión. Tenemos entonces dos tipos de encapsulado que tiene lugar en los diversos archivos y procesos VirtualBox:

1. Arquitectura cliente/servidor. Todos los aspectos de VirtualBox y las máquinas virtuales que están corriendo pueden ser controlados con una simple y aun así poderosa, COM/XPCOM API. Por ejemplo, existe una utilidad de línea de comando llamada VBoxManage que permite controlar las máquinas virtuales tal como lo hace la interfaz gráfica.
2. Arquitectura Frontend/Backend. Las entrañas de VirtualBox (todo eso que hace la virtualización x86 complicada) están ocultos en una librería compartida, VBoxVMM.dll, o VBoxVMM.so en Linux. Esto pude ser considerado un "backend" o caja negra, la cual es estática y es relativamente fácil escribir otro "frontend" sin tener que lidiar con la complejidad de la virtualización x86.

Dentro de una máquina virtual

Desde la perspectiva del SO anfitrión, una máquina virtual es simplemente otro proceso. El SO anfitrión no necesita muchas modificaciones para soportar la virtualización. Aun pensando que hay un driver corriendo en el anillo 0 (recuerde el modelo de la cebolla) que debe cargarse en el SO anfitrión para que funcione VirtualBox, este driver hace menos de lo que usted puede pensar. Sólo se necesita para unas pocas tareas específicas, tales como:

- Asignación de la memoria física para la VM;
- Guardar y recuperar los registros del CPU y tablas de descriptores cuando ocurre una interrupción del anfitrión mientras se ejecuta código de un invitado en el anillo 3 (por ejemplo, cuando el SO anfitrión OS quiere reorganizar la planificación de tareas);
- Al cambiar de contexto desde el anfitrión en anillo 3 al invitado;
- Habilitar o deshabilitar el soporte para VT-x etc. (procesadores avanzados).

Lo más importante, el driver de anillo 0 del anfitrión no interfiere con el planificador de tareas y prioridades de su SO. El SO invitado entero, incluyendo sus propios procesos, es programado para ejecutarse (sheduled) sólo cuando el SO anfitrión le asigna al proceso de la VM un tiempo de ejecución (timeslice) de acuerdo con su prioridad como una tarea más.

Luego de que una VM se ha iniciado, desde el punto de vista de su procesador, su computadora puede estar en uno entre varios estados (lo siguiente requerirá una buena comprensión de la arquitectura de anillo del x86):

1. Su CPU puede estar **ejecutando código de anfitrión de anillo 3** el (por ejemplo de otros procesos del anfitrión), o **código de anfitrión de anillo 0**, como sería si VirtualBox no estuviese ejecutándose.
2. Su CPU puede estar **emulando el código del invitado** (dentro del proceso VM de anillo 3 del anfitrión). Básicamente, VirtualBox intenta ejecutar nativamente tanto código del invitado como posible. Pero puede (más lentamente) emular el código del invitado como recurso cuando no está seguro de lo que el sistema del invitado está haciendo, o cuando la penalización de performance de la emulación no es demasiado alta. Nuestro emulador (en src/emulator /) está basado en QEMU y típicamente funciona cuando:
    - el código del invitado desactiva las interrupciones y VirtualBox no pueden deducir cuando se habilitarán nuevamente (en estas situaciones, VirtualBox

analiza el código del invitado usando su propio desensamblador en src/VBox/Disassembler/);
    - para la ejecución de ciertas instrucciones individuales; esto pasa típicamente cuando una instrucción privilegiada del invitado como LIDT ha causado una excepción y ha necesitado ser emulada;
    - para cualquier código de modo real (por ejemplo, código de BIOS, un invitado de DOS, o cualquier inicio del sistema operativo).
3. Su CPU puede estar **ejecutando nativamente código de invitado de anillo 3** (dentro del proceso VM de anillo 3 del anfitrión). Con VirtualBox, nosotros llamamos a esto "anillo 3 crudo". Esto es, claro, la manera más eficaz de ejecutar al invitado, y en lo posible nosotros no dejamos a menudo este modo.
4. Su CPU puede estar **ejecutando nativamente código de invitado de anillo 0**. Aquí es donde las cosas se ponen difíciles: el invitado sólo piensa que está corriendo código en anillo 0, pero VirtualBox ha engañado al OS invitado para entrar en anillo 1 (que normalmente no es usado en los sistemas operativos x86).

También, en el código fuente de VirtualBox, usted encontrará las muchas referencias a "contexto del anfitrión" o "contexto del invitado". Esencialmente, éstos significan:

- **Contexto del anfitrión (HC**) significa que el OS anfitrión tiene el control de todo, incluyendo la memoria virtual. En las fuentes de VirtualBox, el término "HC" normalmente se referirá sólo al contexto de anillo-3 del anfitrión. Nosotros sólo usamos al contexto de anillo-0 del anfitrión (R0) con nuestro nuevo soporte para Intel VT-x (Vanderpool), por lo que dejaremos el tema para después.
- **Contexto del invitado (GC)** significa básicamente que OS invitado tiene el control, pero con VirtualBox observando en las cosas en segundo plano. Aquí, VirtualBox ha preparado el CPU y la memoria exactamente la manera que el invitado espera, pero se ha insertado "al fondo de la foto". Puede asumir el mando entonces cuando ocurren cosas no permitidas. (una instrucción privilegiada se ejecuta, las excepciones del invitado o las interrupciones externas ocurren). VirtualBox puede delegar entonces el manejo las tales cosas al SO anfitrión. Así que, en el contexto del invitado, nosotros tenemos:
    - anillo 3 (es de esperar que sea ejecutado en "modo crudo" todo el tiempo);
    - anillo 1 (el invitado piensa que es anillo 0, vea lo anterior), y
    - anillo 0 (el cual es código VirtualBox). Este contexto de código de invitado de anillo 0, también se llama a menudo "**hypervisor**".

Soporte de Intel VT-x (" Vanderpool") y AMD-V (SVM)

Con sus últimos procesadores, Intel ha introducido el soporte de virtualización por hardware, llamado "Vanderpool", "IVT", "VT-x", o "VMX" (para las "extensiones de máquina virtual"). Cuando VirtualBox comenzó hace bastante con esto, usaba internamente el término "VMX". Una explicación completa de esta arquitectura puede encontrarse en las páginas de Ref (1), pero con estas extensiones, un procesador opera siempre en uno de los dos modos siguientes:

- En el **modo raíz**, su comportamiento es muy similar al modo normal de funcionamiento (sin VMX), y éste es el contexto que se ejecuta un monitor de la máquina virtual (VMM).
- El modo **no-raíz** (o contexto de invitado) se diseñó para ejecutar una máquina virtual.

Una novedad notable es que los cuatro niveles de privilegio (anillos) son soportados en cualquier modo, de manera que el software de invitado puede correr teóricamente en cualquiera de ellos. VT-x define entonces las transiciones del modo raíz al modo no-raíz (y viceversa) y llama a éstos "entrada de VM" (VM entry) y "salida de VM" (VM exit).

En el modo no-raíz, el procesador causará automáticamente VM exits para ciertas instrucciones privilegiadas y eventos. Para algunas de estas instrucciones, es configurable de qué manera deben ocurrir las salidas de VM.

Como dijimos anteriormente, nuestro hypervisor, en máquinas no-VT-x, está en el anillo 0 del contexto de invitado, debajo del código de invitado anillo-0 (que realmente se ejecuta en anillo 1). Cuando VT-x se habilita, el hypervisor puede estar seguro en el contexto de anillo 0 del anfitrión y se pueden activar automáticamente por el uso de las nuevas salidas de VM.

VirtualBox también soporta totalmente AMD-V, SVM de AMD y VT-x de Intel.

Referencia (1):
http://www.virtualbox.org/wiki/VirtualBox_architecture (traducción: FFP)

VirtualBox – Ejemplo de aplicación

A continuación, se puede ver una sesión de VirtualBox ejecutándose en Ubuntu Linux. La configuración del programa es muy simple. En este caso se creó un disco rígido virtual de 30 MB para instalarle DOS 6.22.

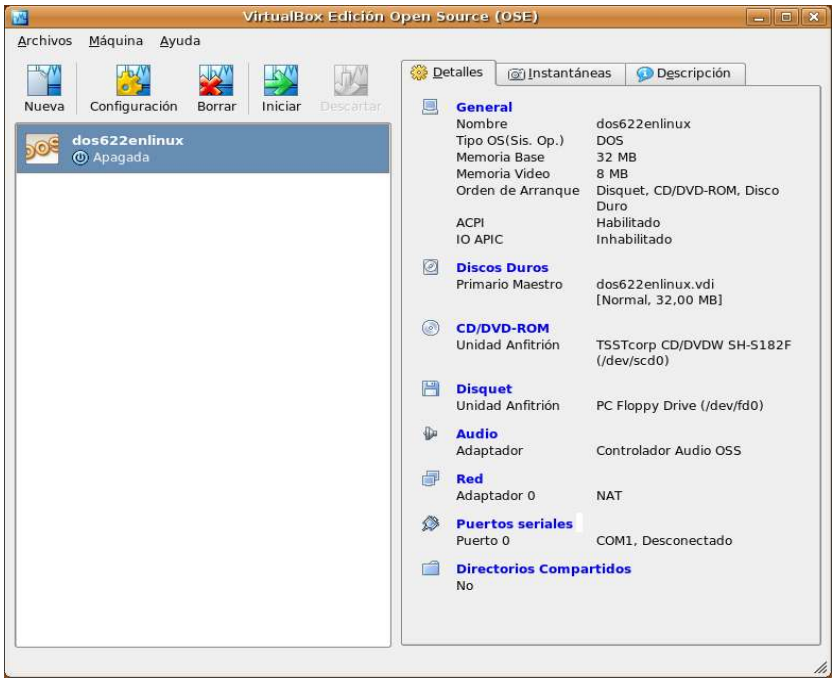


Fig. 1.9.2 – Sesión de VirtualBox

La máquina virtual se arranca con el botón Iniciar. Aparece una nueva ventana que muestra el proceso de "booteo". En este caso se inició desde el CD ROM de Técnicas Digitales III, mostrando su menú principal.

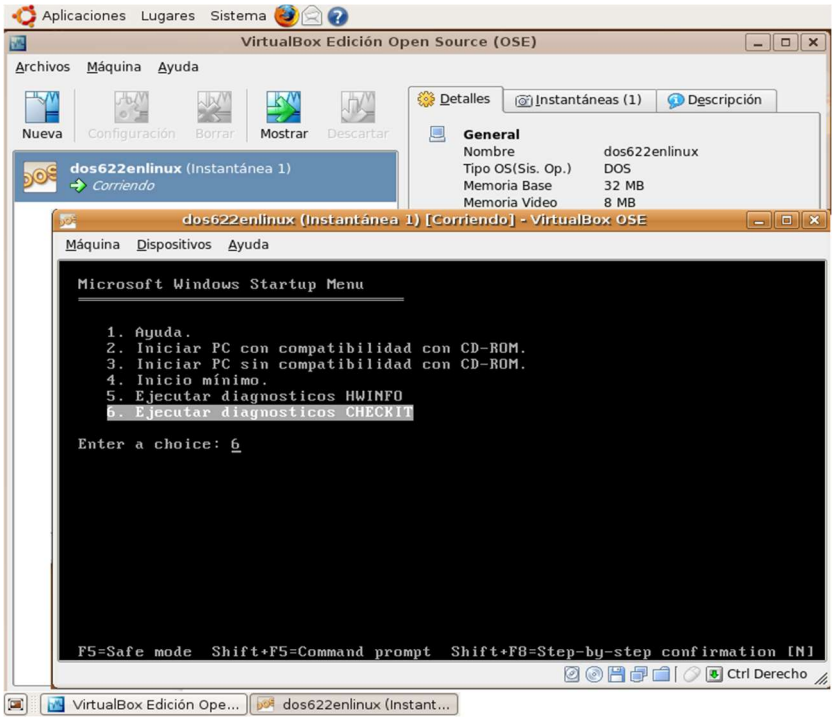


Fig. 1.9.3 – Booteo de VirtualBox

1.9.2 VMware - características

Cuando se desarrolla software que se ejecutará con otro sistema operativo, por ejemplo Windows, Linux o Novel Netware, las distintas máquinas virtuales ven la CPU, la memoria, la placa de red y los discos como si fuesen propios.

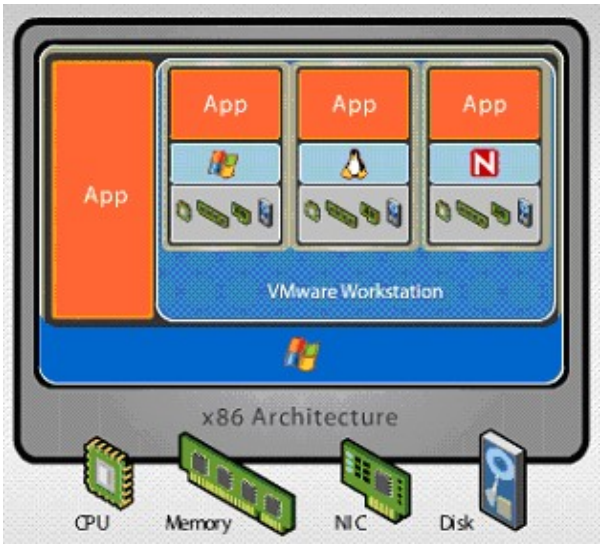


Fig. 1.9.4 – Arquitectura de VMware – Sistemas operativos

También cuando se está desarrollando una aplicación que se ejecutará en red. En este caso, las distintas máquinas virtuales se podrán comunicar entre sí como si fuesen reales.

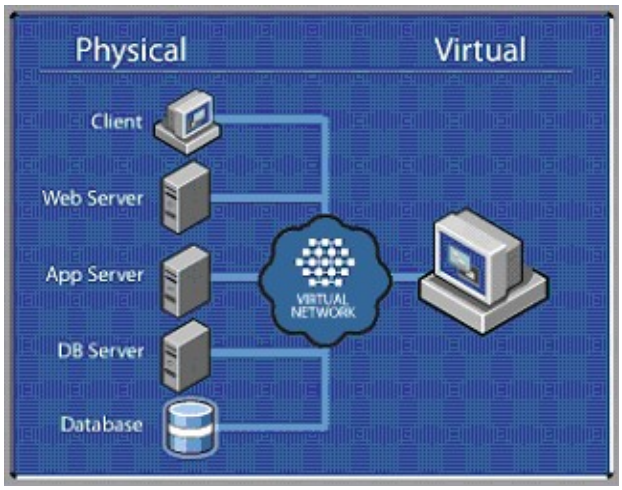


Fig. 1.9.5 – Arquitectura de VMware

Por último, por tratarse de una máquina virtual, se la puede iniciar (CTL-ALT-DEL), se puede particionar su disco rígido, se lo puede formatear, etc., sin afectar a la máquina ni al sistema operativo anfitrión ni a las demás máquinas virtuales.

Si la máquina virtual se ve infectada por virus informáticos, la máquina anfitrión no será afectada.

Desde los sitios Web de Microsoft, VMware y VirtualBox se pueden descargar versiones de estos programas según el SO que actúe como anfitrión (host).

1.10 MULTIPROCESAMIENTO

1.10.1 Introducción

Multiprocesamiento significa usar en un sistema dos o más procesadores coordinados entre sí. El multiprocesamiento se ha ido tornando más atractivo a medida que el costo de los procesadores ha ido bajando. La performance se puede incrementar substancialmente distribuyendo las tareas del sistema entre procesadores separados que las ejecuten concurrentemente. Además, el multiprocesamiento incita a un diseño modular, lo que redunda usualmente en sistemas que son más fáciles de mantener y ampliar.

En general, el uso de procesadores múltiples ofrece varias ventajas significativas respecto del método centralizado, que depende de una sola CPU y una memoria extremadamente rápida:

- Las tareas del sistema pueden ser asignadas a procesadores específicos cuyos diseños están optimizados para realizar dichas tareas de manera fácil y eficiente.
- Se pueden lograr muy altos niveles de performance cuando los procesadores pueden ejecutar simultáneamente (procesamiento distribuido/paralelo).
- La subdivisión del sistema estimula el desarrollo en paralelo de los subsistemas, divide la aplicación en tareas más pequeñas y por ende más manejables, y ayuda a aislar los efectos de modificaciones en el sistema.

Existen dos tipos de procesadores: los procesadores independientes y las extensiones del procesador, también llamadas coprocesadores. Un procesador independiente ejecuta su propia secuencia de instrucciones. El 8086, el 8088 y el 8089 son ejemplos de procesadores independientes. Un segundo tipo de procesador, llamado "extensión del procesador", tal como el 8087 NPX, agrega registros, tipos de datos y nuevas instrucciones al sistema. Una extensión del procesador, en efecto, extiende el conjunto de instrucciones (y la arquitectura) de su procesador principal.

En los sistemas de multiprocesamiento existen dos problemas de coordinación clásicos: arbitraje del bus y exclusión mutua. El arbitraje del bus puede ser realizado por la lógica de pedido/concesión del bus contenida en cada uno de los procesadores (arbitraje de bus local), por árbitros de bus (arbitraje del bus del sistema) o por una combinación de ambos cuando los procesadores comparten múltiples buses.

A medida que cada CPU requiere asincrónicamente el acceso al bus compartido, la lógica de arbitraje resuelve prioridades y concede acceso al bus, la CPU completa su transferencia y luego cede el bus o espera ser forzada a ceder el bus. En todos los casos, el arbitraje opera invisible al software. Para la exclusión mutua, cada procesador tiene una señal de bloqueo para el bus (bus LOCK), que puede ser activada por un programa para prevenir que otros procesadores obtengan acceso al bus compartido. Cada procesador tiene una instrucción que puede examinar un byte de memoria mientras el bus está bloqueado, permitiendo de esta manera implementar un mecanismo de señalización (semáforo) para controlar el acceso a los recursos compartidos (tablas, buffers, etc.).

Nota: el desarrollo precedente corresponde al inicio del multiprocesamiento por parte de Intel y su implementación a fines de los 90. Los procesadores actuales continúan con el mismo concepto con la salvedad de que el aumento de la integración posibilitó incorporar en un mismo chip varios procesadores, las unidades de punto flotante y el I/O necesario. El aumento de la cantidad de pines evita el multiplexado con el consiguiente aumento de performance en el bus.

1.10.2 Arquitectura Multibus

Para soportar la interfaz de múltiple maestro para la familia 8086/88, Intel incluyó el árbitro de bus 8289. El 8289 es compatible con el bus local del 8086 y en conjunto con el controlador de bus 8288 implementa el arbitraje del bus con el protocolo MULTIBUS.

Véase que en la figura se representan 2 procesadores idénticos que se comunican entre sí sólo por el bus y los recursos compartidos, de manera que si fuese necesario colocar mas procesadores se pueden agregar en paralelo sin mayores inconvenientes.

La mayoría de los procesadores actuales no se pueden conectar de esta forma, ya que la estrategia de los fabricantes ha cambiado, promoviendo el cambio de modelo y no la ampliación mediante multiprocesamiento.

Los procesadores Intel aptos para multiprocesamiento son los de la línea Xeon, destinados principalmente a servidores.

Adicionalmente, los procesadores actuales son multinúcleos, con lo que el multiprocesamiento se realiza en el interior del chip y el SO distribuye las tareas a los distintos núcleos

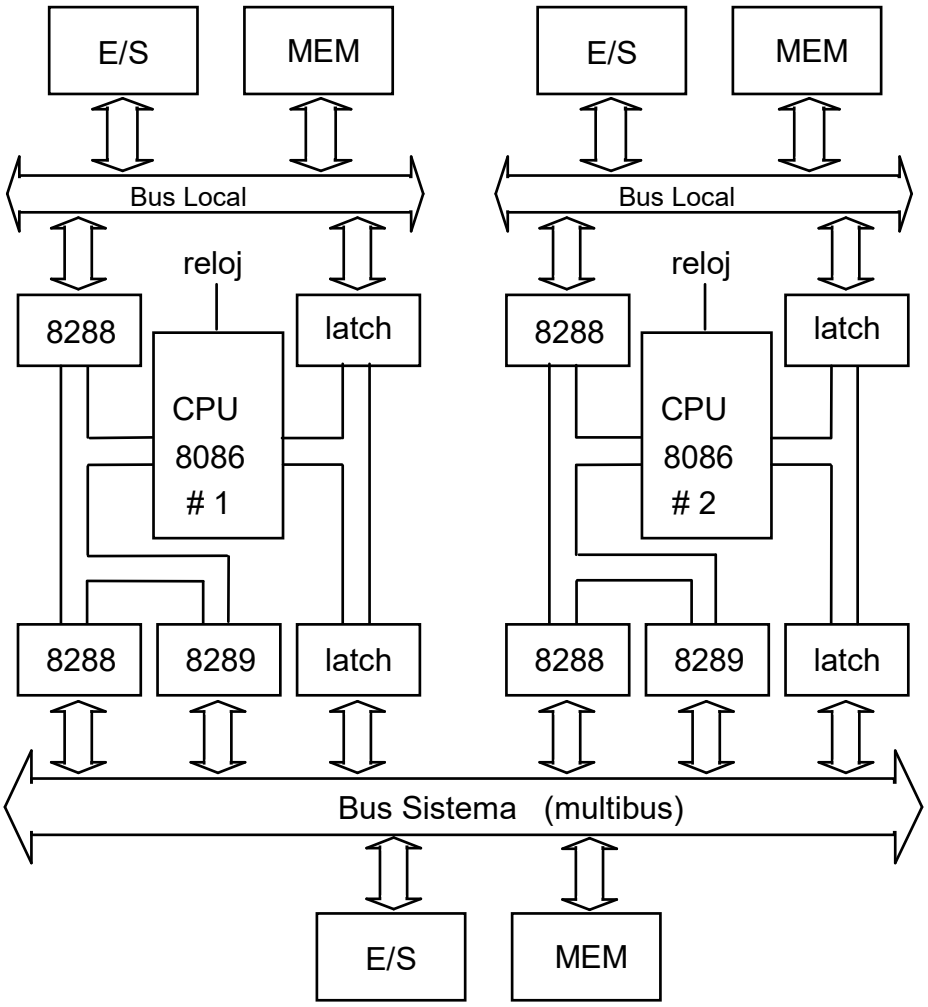


Fig. 1.10.1 - Sistema de dos procesadores usando Intel 8086

1.10.3 Ejemplo – problema del semáforo

Se trata de un sistema de dos procesadores conectados como indica la figura 1.10.1. El primero de ellos (P1) adquiere datos de 5 temperaturas de un proceso cada 30 segundos. Luego de realizar la conversión a ºC los debe enviar al otro procesador (P2) para realice los cálculos apropiados de acuerdo con un modelo matemático y evalúe las correcciones necesarias.

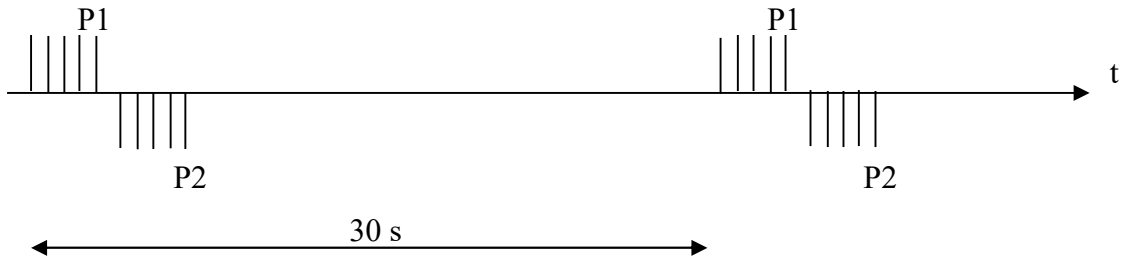


Fig. 1.10.2 – Diagrama de tiempos del ejemplo

Para comunicar los procesadores se desea utilizar una tabla de 5 valores en la memoria compartida o de sistema (ver Tabla 1.10.1). T1 (0) indica el valor de temperatura 1 adquirido en el muestreo 0. La dirección del primer dato es acordada entre los dos procesadores y las de los restantes siguen en orden consecutivo. El procesador P2 debe poder leer la tabla en forma asincrónica, es decir, cuando lo estime conveniente, sin previo aviso de P1. Los 5 valores deben ser considerados como un "conjunto" de datos, que sólo es válido cuando hayan sido obtenidos en el mismo ciclo de muestreo.

El inconveniente que se puede presentar es que P2 lea la tabla de datos mientras P1 la está actualizando. En este caso, el conjunto de valores será una mezcla de datos nuevos, adquiridos en el muestreo (1) y viejos (0) (ver Tabla 1.10.2).

T1 (0)
T2 (0)
T3 (0)
T4 (0)
T5 (0)

Tabla 1.10.1

T1 (1)
T2 (1)
T3 (1)
T4 (0)
T5 (0)

Tabla 1.10.2

Para intentar la solución de este problema aprovecharemos las facilidades de HW y SW de ambos procesadores y agregaremos una posición más en la tabla, que será una bandera o semáforo para controlar el acceso a la tabla. La misma en 00 indica que el conjunto de datos es coherente y corresponde al mismo muestreo. En este estado puede ser leído por P2. Con criterio más amplio, la tabla está "habilitada". La bandera en FF indica que la tabla está siendo actualizada por P1 o leída por P2. En este estado, podemos decir que la tabla está "bloqueada" por el procesador que la está accediendo.

Partimos de la situación en que los datos son coherentes y FLAG=00. Cuando P2 decide leer, verifica primero si FLAG=00. Si es así, pone FLAG=FF y luego lee los 5 datos. Si en el

primer intento FLAG=FF, espera e intenta nuevamente. Luego de leer todos los datos, P2 pone FLAG=00. Cuando P1 necesita actualizar los datos, verifica primero si FLAG=00. Si es así, pone FLAG=FF, escribe los 5 datos y pone FLAG=00.

FLAG (00)
T1 (0)
T2 (0)
T3 (0)
T4 (0)
T5 (0)

Tabla 1.10.3 (habilitada)

FLAG (FF)
T1 (1)
T2 (1)
T3 (1)
T4 (0)
T5 (0)

Tabla 1.10.4 (bloqueada)

Parecería que con este simple artilugio de SW es suficiente para solucionar nuestro problema, sin embargo, no es así. En la Tabla 1.10.5 analizamos las actividades de P1 y P2 en el tiempo (secuencia de ciclos de bus: 1, 2, 3...). En la columna FLAG está el valor "verdadero" de esa posición de memoria de sistema. En P1 está la "copia" que hace P1 cuando lee la memoria de sistema y la lleva a un registro interno donde determinará su valor (00 o FF) y las instrucciones ejecutadas. Ídem P2 para el procesador P2. Obsérvese que P1 y P2 no pueden utilizar el bus al mismo tiempo.

Tiempo	FLAG	P1	P2
1	00	CMP, FLAG=00	-
2	00	JMPE	CMP, FLAG=00
3	FF	MOV, FLAG=FF	JMPE
4	FF	Conversión T1	MOV, FLAG=FF
5	FF	ídem	Preparación lectura
6	FF	ídem	ídem
7	FF	MOV, P1→T1	ídem
8	FF	Conversión T2	MOV, T1→P2
9	FF	ídem	-
10	FF	ídem	MOV, T2→P2
11	FF	MOV, P1→T2	-
12	FF	Conversión T3	MOV, T3→P2
13	FF	ídem	-
14	FF	ídem	MOV, T4→P2
15	FF	MOV, P1→T3	-
16	FF	Conversión T4	MOV, T5→P2

Tabla 1.10.5

Como se puede apreciar, en el tiempo 16 el procesador P2 leyó los 5 valores, pero T1 corresponde a una muestra distinta (más nueva) que los restantes (T2 a T5). Esto ocurrió porque los dos procesadores "vieron" el FLAG en 00 y por lo tanto habilitada para operar.

Tiempo	**FLAG**	**P1**	**P2**
1	00	MOV AL,FF	-
2	FF	LOCK XCHG FLAG=FF	MOV AL,FF
3	FF	TEST AL,AL FLAG=00	LOCK XCHG FLAG=FF
4	FF	Habilitado para escribir	TEST AL,AL FLAG=FF
5	FF	Conversión T1	Bloqueado para leer
6	FF	-	-
7	FF	-	-
8	FF	MOV, P1→T1	-
9	FF	Conversión T2	-
10	FF	-	-
-	-	-	-
24	FF	MOV, P5→T5	-
25	00	MOV FLAG,00	-
26	00	-	MOV AL,FF
27	FF	-	LOCK XCHG FLAG=00
28	FF	-	TEST AL,AL FLAG=00
29	FF	MOV AL,FF	Habilitado para leer
30	FF	LOCK XCHG FLAG=FF	-
31	FF	TEST AL,AL FLAG=FF	MOV, T1→P2
32	FF	Bloqueado para escribir	-
33	FF	-	MOV, T2→P2
-	-	-	-
39	FF	-	MOV, T5→P2
40	FF	-	MOV, 00→FLAG

Tabla 1.10.6

Para solucionar esto, los procesadores como el Intel 8086, diseñado para trabajar en sistemas de procesadores múltiples, cuenta con un prefijo (instrucción LOCK) para bloquear el bus con la señal LOCK mientras determina el estado de la bandera y una instrucción XCHG para intercambiar un registro y memoria. De esta manera, el otro procesador no puede acceder a la tabla ni a la bandera hasta que el primero no haya fijado "su" bandera y por tanto asegurado la integridad de los datos. El Pentium agrega la instrucción CMPXCHG8B *reg*, *mem64* (*Compare and Exchange 8 Bytes*) que hace lo siguiente: compara el valor de 64 bits ubicado en EDX:EAX con un valor de 64 bits situado en memoria. Si son iguales, el valor en memoria se reemplaza por el contenido de ECX:EBX y el indicador ZF se fija en 1. En caso contrario, el valor en memoria se carga en EDX:EAX y el indicador ZF se fija en cero.

Volviendo al problema, la tabla de tiempos quedaría modificada como muestra la Tabla 1.10.6, donde se ve que ahora funciona correctamente.

Nota: se puede ver una animación paso a paso de este ejemplo, en la página web de Técnicas Digitales III (http://www.frsn.utn.edu.ar/tecnicas3).

1.11 COPROCESADORES

1.11.1 Introducción

En la figura se puede ver el esquema de interconexión entre la CPU Intel 8086, el coprocesador matemático 8087 y los circuitos auxiliares para su correcta operación.

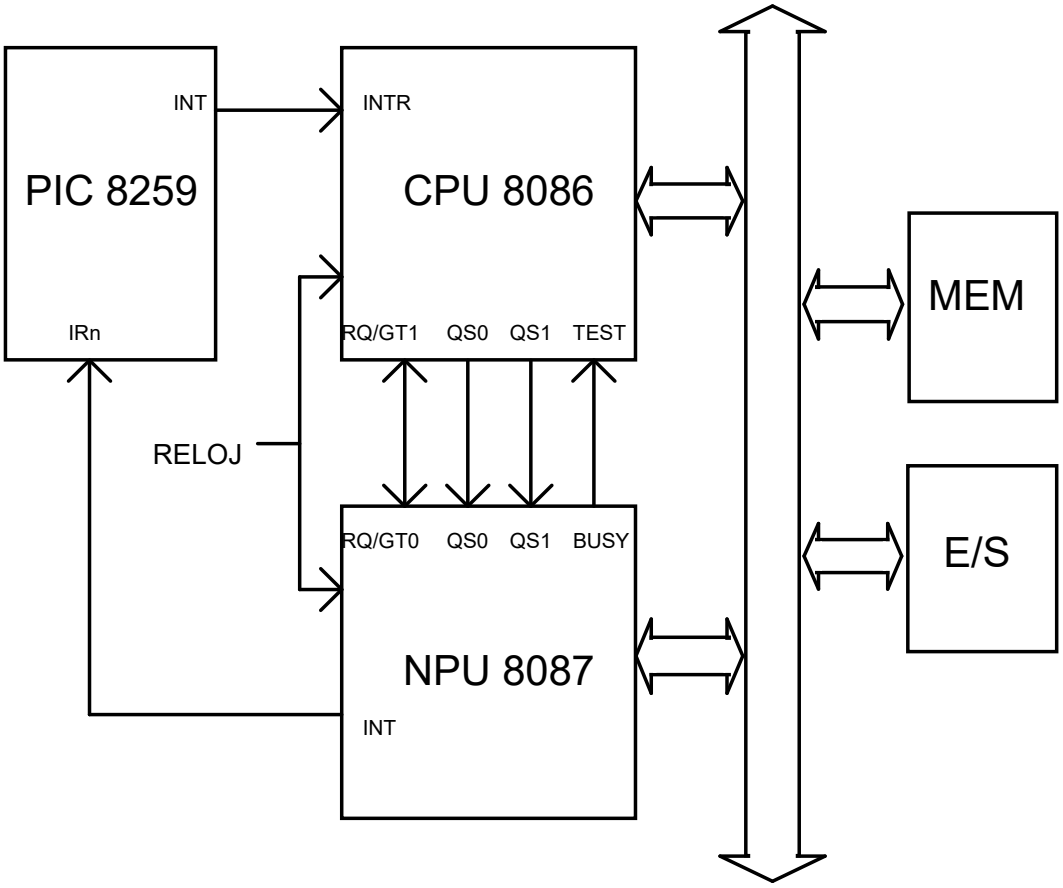


Fig. 1.11.1 - Coprocesador matemático Intel 8087

El 8087 NPX (extensión de procesamiento numérico) realiza operaciones aritméticas y de comparación sobre una variedad de tipos de datos; también ejecuta numerosas funciones trascendentes (ej.: tangente y logaritmo). El programador generalmente no percibe al 8087 como un dispositivo separado; en cambio, la capacidad de cálculo de la CPU aparece enormemente expandida.

A partir del 80486 y Pentium todos los procesadores ya incorporan el coprocesador en el mismo encapsulado, no obstante, todos los conceptos mencionados se mantienen por la compatibilidad hacia atrás que mantienen Intel y AMD en sus productos.

Para hacerse una idea de la mejora que puede se obtener, en el siguiente cuadro se comparan los tiempos de ejecución (en µS) del 8087 y el 8086 para realizar las mismas funciones.

Instrucción	*8087*	*8086*
Multiplicación	19	1600
Suma	17	1600
División	39	3200
Comparación	9	1300
Raíz cuadrada	36	19600
Tangente	90	13000

Tabla 1.11.1 - Coprocesador matemático 8087

1.11.2 Aplicaciones

- *Procesamiento de datos comercial.*
- *Robótica.*
- *Navegación.*
- *Terminales gráficas.*
- *Adquisición de datos.*
- *Control de procesos.*
- *Control numérico.*

Una instrucción numérica para el 8087 aparece como una instrucción ESCAPE para el 8086; ambos descodifican y ejecutan la misma instrucción simultáneamente. Si la instrucción ESCAPE hace referencia a memoria, la CPU direcciona y luego hace una lectura simulada, ignorando el dato, pero el NDP toma el dato y la dirección. Si el NDP requiere más bytes, gestiona el bus vía RQ/GT y con la dirección capturada previamente lee todos los datos que necesita. En siguiente ejemplo, la CPU (bloques inferiores) detecta ESCAPE simultáneamente al NDP, el cual comienza el cálculo apenas tiene los datos. La CPU continúa ejecutando instrucciones en paralelo hasta que necesita los resultados del NDP. En este punto espera (WAIT) chequeando la línea TEST el final del cálculo.

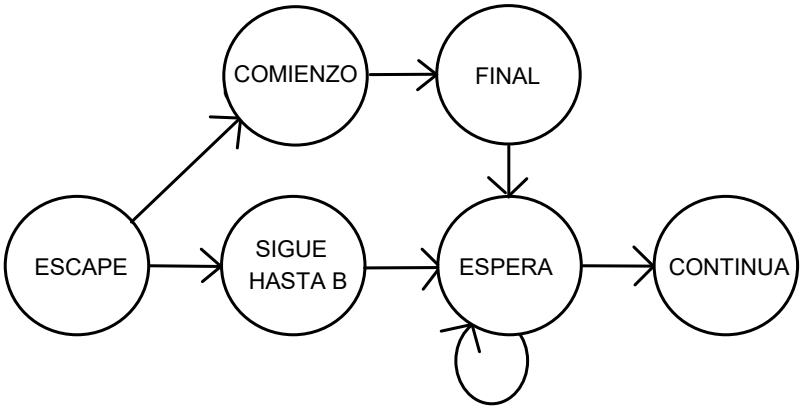


Fig. 1.11.2 – Flujo de operaciones del coprocesador matemático

En los procesadores 80486, Pentium y posteriores el coprocesador matemático se encuentra en el mismo encapsulado que el procesador y para el programador resulta simplemente un procesador con más instrucciones.

CAPÍTULO 2

PROGRAMACIÓN EN AMBIENTE DE PC

2.1 EL SISTEMA OPERATIVO

El sistema operativo (SO) provee el ambiente para que otros programas se ejecuten. En las computadoras que tienen múltiples usuarios, todos compiten por los recursos de la máquina. El sistema operativo es el árbitro que decide que determinado usuario acceda a determinado recurso y en que momento. El sistema operativo previene que un usuario interfiera con otro. El sistema operativo también ofrece un conjunto de servicios que libera al usuario de lidiar con las dificultades de hardware. Esos servicios mantienen al usuario advertido si pretende realizar operaciones que podrían destruir datos o programas de otros usuarios. Las necesidades de la mayoría de los usuarios reducen las libertades individuales de los programas.

2.1.1 Clasificación

Podemos clasificar a los sistemas operativos en multi-usuarios y mono-usuarios. Los sistemas operativos multi-usuarios deben ejecutarse en computadoras que contengan el hardware que soporte los niveles de privilegio y la protección de memoria de manera que se mantenga la aislación adecuada entre los distintos usuarios. Como ejemplos podemos citar el Windows (2000, NT, XP, Vista, 7 y posteriores), el UNIX y el Linux.

También los podemos considerar como sistemas en tiempo real o no. Los sistemas en tiempo real son los que proveen un servicio en un lapso adecuado a la necesidad del proceso que lo requiere. Tal es el caso de las computadoras de control de procesos. Como ejemplo podemos citar el QNX.

Otra distinción puede ser la interfaz. Están los de acceso por línea de comandos, o texto (DOS) y los de interfaz visual o gráfica (Windows, Linux). En algunos casos la interfaz visual está por encima de un sistema de línea de comandos pero pueden separarse claramente las funciones de cada uno (Linux, Windows 95).

Por último, tenemos los de código abierto (open source) y los privativos. En los primeros está disponible al público el programa fuente, lo que permite revisiones y mejoras soportadas por una comunidad de programadores (Linux), a diferencia de los otros, donde las actualizaciones las hacen las grandes empresas de software (Microsoft).

2.1.2 Programación del SO

Los primeros sistemas operativos se programaron en lenguaje assembly, ya que era lo natural para una interfaz entre el procesador (hardware) y el usuario. Cada instrucción en lenguaje assembly se corresponde con una instrucción del lenguaje de máquina. Esto tenía la ventaja de la eficiencia en el uso de los recursos del harware pero la desventaja de tener que re-escribir todo el código si cambiaba el hardware.

El lenguaje de programación C fue diseñado e implementado en ambiente UNIX, con el objetivo de achicar la "brecha semántica" entre el assembly y el usuario, pero tratando de mantener expresiones breves. Con el tiempo sobrevivió a una cantidad de competidores, siendo hoy uno de los mas usados para crear aplicaciones eficientes y la base de los sistemas UNIX y sus derivados, como Linux. se dice que estos sistemas tienen un 95% de su código en C y sólo el 5% restante en assembly, lo que facilita su adaptación al harware en constante evolución.

2.1.3 Sistema operativo DOS

Para las pequeñas computadoras, como fue en sus comienzos la IBM PC, un sistema operativo mono-usuario era suficiente. Las PC eran utilizadas por una sola persona a la vez. A nivel de programa, no hay prohibición en contra de realizar cualquier secuencia de instrucciones. El propósito de sistema operativo DOS de las PC era proveer un ambiente para operar y un conjunto de servicios al usuario.

El usuario puede ser un programador o puede ser algún otro programa de aplicación. Por ejemplo, cuando usted se sienta frente al teclado y comienza a trabajar con el sistema, usted es un usuario del DOS. Pero cuando el editor salva un archivo en diskette, el editor usa el DOS. El editor está usando los servicios del DOS para salvar el archivo, en lugar de incluir los programas para hacer tal trabajo en el editor.

El servicio principal del DOS en una PC es proveer un sistema de archivos y un ambiente de ejecución para los programas. El sistema de archivos es el método por el cual son salvados y recuperados los datos en diskettes o discos rígidos. Si todos los programas de aplicación usan el DOS para salvar información, entonces ellos pueden compartir dicha información.

2.1.4 Sistema de archivos

Es un conjunto de datos relacionados de alguna manera. El creador del archivo le asigna un nombre. El archivo está compuesto de registros (records). El registro es un conjunto de bytes. El programador define el sentido de los bytes en el registro.

Un programa en assembly puede almacenarse como un archivo con un nombre (PROGRAMA.ASM). El archivo está compuesto de registros, donde cada registro es una única sentencia en lenguaje assembly. Cada registro tiene una composición que tiene significado para el programador y no para el DOS. Las diferentes áreas en cada registro son los campos. Al DOS no le interesa como se dividen los registros en campos; este trabajo se deja para la aplicación, en este caso el ensamblador.

Nombres de archivos

El nombre puede tener de uno a ocho caracteres (NOMBRE.EXT). La extensión, de uno a tres caracteres, indica el tipo de archivo. En el caso del ensamblador, existe un archivo de entrada, con extensión .ASM y tres archivos de salida con extensiones: .OBJ para el objeto, .LST para el listado y .CRF para las referencias cruzadas.

Directorios o carpetas

El sistema operativo maneja el método para almacenar múltiples archivos en el mismo disco. El directorio es una tabla de contenidos, que además del nombre del archivo contiene punteros para localizar la posición física de los archivos en el disco y también la fecha de creación. Si hay mas de un disco, el archivo se nombra agregando la identificación de la unidad de disco donde está instalado (ej.: A:NOMBRE.EXT).

2.1.5 Procesador de comandos

El DOS provee un ambiente para ejecutar los programas de aplicación. La primera parte del DOS que el usuario ve es el procesador de comandos, que responde a los comandos del usuario y comienza la ejecución de los programas de aplicación.

Cuando la PC se enciende, se ejecuta una rutina en ROM que prueba los componentes del sistema (Test de encendido o POST) e inicializa los dispositivos de entrada/salida. El resto del sistema básico de entrada/salida (BIOS) que está en la ROM provee al programador en assembly

un conjunto de servicios para acceder a los dispositivos sin especificar la implementación específica del hardware.

Luego de ejecutar el POST, se carga desde el disco una rutina que será luego la encargada de cargar el resto del sistema operativo (BOOT). Cuando este proceso termina, aparece en pantalla el título y nivel de revisión del DOS y luego el aviso (prompt) A> que indica que el procesador de comandos, el sistema de archivos y otros utilitarios han sido cargados y están listos para correr.

El aviso A> tiene dos significados: el > indica que el intérprete de comandos está esperando un comando y la A indica la unidad de discos que se toma por omisión (default drive). Usualmente A y B se asignan a unidades de diskettes y C a un disco rígido.

Hay comandos incluidos o residentes (built-in) que están siempre disponibles y otros que requieren la existencia de un archivo en disco para su ejecución. El usuario teclea el comando, por ejemplo DIR y el intérprete de comandos pasa el control a la rutina correspondiente del DOS, que realiza la función solicitada y devuelve el control al DOS.

Si el usuario ingresa un comando no residente, el intérprete de comandos intenta cargar la rutina correspondiente desde el disco; en este caso el intérprete de comandos actúa como cargador de programas. El intérprete asume que el nombre del comando es el nombre de un archivo, que buscará en el disco con la extensión .COM o .EXE. Estos son archivos de comando y archivos ejecutables, que están en lenguaje de máquina.

También se pueden invocar para su ejecución los archivos de comandos indirectos (batch), que tienen una extensión .BAT y son archivos de texto que contienen comandos que interpreta el procesador de comandos como si ingresaran por teclado. Hay un archivo de comando especial, llamado AUTOEXEC.BAT que se ejecuta, si existe, inmediatamente luego que el DOS es cargado. Se utiliza para arrancar la computadora de alguna manera particular con programas de aplicación.

2.1.6 Funciones del DOS

A diferencia de los comandos, las funciones son llamadas desde los programas de aplicación, en lenguaje assembly. Un programa ejecuta una función del DOS a través de una interrupción de software, sin saber en qué posición de memoria se encuentra la rutina de atención de esa función. Los vectores de interrupción son cargados por el DOS durante la inicialización del sistema. Una misma interrupción puede realizar distintas funciones, cargando distintos valores en el registro AH antes de ejecutar la instrucción INT.

2.2 CREACIÓN DE UN PROGRAMA EN LENGUAJE ASSEMBLY

2.2.1 Editor

En primer lugar, el EDITOR crea un programa fuente en formato texto. Éste puede ser en lenguaje assembly o lenguaje C. Este programa se almacena en un archivo, generalmente con la extensión .ASM o .C.

Se puede usar cualquier editor de texto, dependiendo de la cantidad de memoria y espacio de almacenamiento disponible. Los programas más grandes, en general brindan prestaciones extra, como indentación automática, diferentes colores para instrucciones y nombres de variables, plantillas, integración con otros programas, etc.

El archivo fuente debe respetar la sintaxis normalizada, tanto para las directivas del ensamblador (ver apéndice al final del apunte) como para las instrucciones del procesador donde será ejecutado el programa. Lo mismo aplica para el compilador C.

La figura es un diagrama de flujo que muestra las actividades involucradas cuando se trabaja en lenguaje assembly y C en una computadora personal compatible. Los hexágonos representan programas utilitarios, los círculos indican archivos en disco, generados por el usuario y por último las flechas guían las actividades.

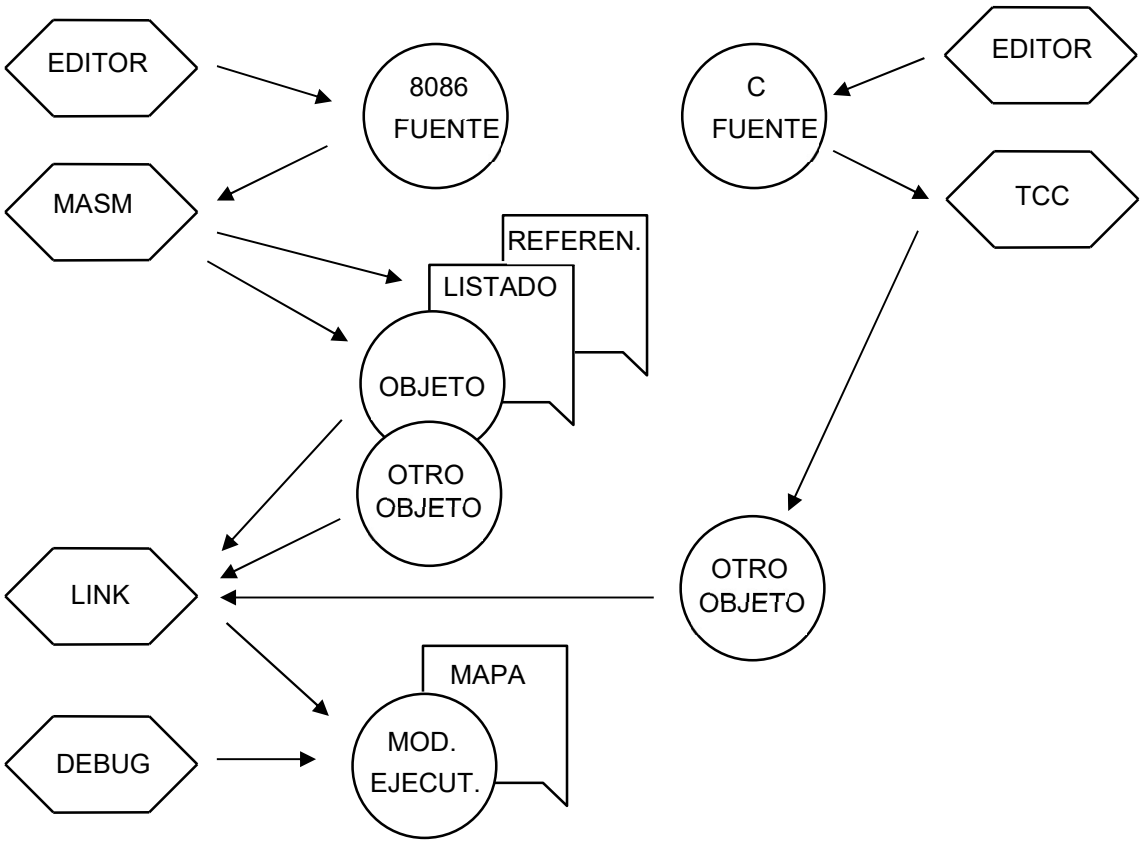


Figura 2.2.1 – Programación en ambiente de PC

2.2.2 Macro ensamblador y turbo ensamblador

Hay dos versiones del ensamblador: el MASM.EXE (macro ensamblador) y el TASM.EXE (turbo ensamblador). Ambos cumplen la misma función, por lo que hablaremos simplemente de ensamblador, estudiándose en la práctica las particularidades de cada uno de ellos.

La entrada al ensamblador es el archivo fuente creado por el editor de textos y está en código ASCII. El ensamblador produce hasta tres archivos de salida. El archivo objeto es la versión en lenguaje de máquina del programa, aunque aún no está lista para ser ejecutada.

El archivo listado es un archivo de texto en ASCII que contiene el tanto la información fuente como la generada por el ensamblador. El archivo de referencias cruzadas, que no está en lenguaje de máquina ni es un texto, contiene información acerca del uso de símbolos y rótulos en

el programa. Al igual que el archivo objeto, el archivo de referencias cruzadas requiere un procesamiento adicional antes de que pueda ser usado.

El ensamblador se activa con un comando DOS:

A>MASM

Luego el ensamblador consulta los nombres de los archivos que intervendrán en el ensamblado. El nombre por omisión del archivo para el programa fuente es .ASM, de manera que no es necesario ingresar la extensión. Si no se ingresa otro nombre, el archivo objeto se denominará igual que el archivo fuente, pero con la extensión .OBJ. Para el listado y las referencias cruzadas, en caso de responder RETURN (o ENTER) se utilizará el archivo NUL, que no puede ser leído, como valor por omisión. Luego del ensamblado se indica si hubo errores de advertencia (warning errors) o errores graves (severe errors) tanto en pantalla como en el listado.

Otro método para invocar al ensamblador es:

A>MASM PROG,,,;

donde PROG es el nombre del archivo fuente y las comas indican que se deben tomar las asignaciones por omisión para los nombres de los otros archivos.

El archivo listado se puede ver con el comando DOS:

A>TYPE PROG.LST

Oprimiendo CONTROL y PrtScrn antes del comando TYPE, se puede enviar el listado a la impresora.

Tabla de símbolos

El archivo listado contiene la tabla de símbolos, siguiendo al programa. La misma muestra los segmentos y símbolos usados en el programa. Los símbolos pueden ser variables, rótulos o números y van acompañados por sus atributos, tipo y valor.

Referencias cruzadas

El archivo de referencias cruzadas producido por el ensamblador no está listo para ser usado. Se debe ejecutar el comando CREF para cambiar el archivo .CRF en un archivo de texto. El comando DOS:

A>CREF PROG,

crea el archivo PROG.REF, que puede ser mostrado con el comando TYPE.

La columna de la izquierda contiene los nombres de las variables y símbolos definidos en el programa. A la derecha siguen un grupo de números enteros que indican la línea de programa donde el símbolo aparece. Si el número es seguido de "#" indica que el símbolo fue definido en esa línea. Esto es útil cuando se sigue la pista de una variable cuyo valor es incorrecto: la tabla proporciona los números de línea que hacen referencia a esa variable. También en el caso en que se modifique una subrutina, la tabla indica las porciones de programa que efectúan llamados a dicha subrutina, permitiendo evaluar los efectos de la modificación en cada tramo del programa.

2.2.3 Vinculación

El archivo PROG.OBJ se transforma en PROG.EXE mediante el comando DOS:

A>LINK PROG.OBJ, PROG.EXE, PROG.MAP,,

2.2.4 Compilación

El archivo PROG_C.C se transforma en PROG_C.OBJ y luego en PROG_C.EXE

A>TCC –PROG_C.EXE PROG_C.C *PROG.OBJ (opcional, para incluir .OBJ desde Assembly)*

2.2.5 Depuración

Para ejecutar los programas en ambiente de prueba se utilizan DEBUG y TD.

A>TD

En la práctica se utilizará el ambiente Turbo, que consta de los programas AE (editor), TASM (ensamblador), TLINK (vinculador), TD (depurador) y TREF (referencias). Este editor fue concebido especialmente para trabajar en un ambiente de desarrollo en assembly.

Como alternativa, si se trabaja en ambiente Windows, se puede usar el Notepad y los ambientes de Microsoft o Borland. Existen también IDE (Integrated Development Environment) como Negatory Assembly Studio que están diseñados para Windows y Linux y son gratuitos.

2.3 EJEMPLOS DE PROGRAMACION EN ASSEMBLY

A continuación, se muestra un ejemplo de programa y subrutina, que luego serán ensamblados y vinculados en la PC.

Los archivos que se mencionan están incluidos en el CD de Técnicas Digitales III y en la página web de la materia (http://www.frsn.utn.edu.ar/tecnicas3).

2.3.1 Contenido del archivo ENSAYO.ASM

```
                              ;     PROGRAMA ENSAYO
                              ;
                              ; Este programa crea una tabla de
                              ; datos y llama la subrutina ORDEN_DESC
                              ; pasando la direccion del primer dato
                              ; y la cantidad de datos por la pila.
                              ;
DATOS SEGMENT                 ; Define un segmento llamado DATOS
N     DW  5                   ; Define la variable N con el valor 5
TABLA DW 1,2,3,4,5            ; Define la variable TABLA con el
                              ; valor 1 y asigna valores a cuatro
                              ; palabras mas
CARTEL1 DB 'TECNICAS DIGITALES III$'
CARTEL2 DB ' - LOS DATOS HAN SIDO ORDENADOS$'
DATOS       ENDS              ; Fin del segmento DATOS
                              ;
PILA SEGMENT STACK            ; Define un segmento PILA de tipo STACK
DW  128 DUP (?)               ; Reserva 128 palabras
CAB_PILA LABEL WORD           ; Nombra CAB_PILA al tope del stack
PILA  ENDS                    ; Fin del segmento PILA
                              ;
PROG SEGMENT PUBLIC           ; Define un segmento PROG de tipo PUBLIC
EXTRN ORDEN_DESC:NEAR         ; Define subrutina externa y proxima
ASSUME CS:PROG,DS:DATOS,SS:PILA; Asocia CS al segmento PROG , DS al
                              ; segmento DATOS y SS al segmento PILA
                              ;
COMENZAR:   MOV AX,DATOS      ; Inicializa el registro DS con el
            MOV DS,AX         ; valor DATOS, obtenido por el MASM
            MOV AX,PILA       ; Inicializa el registro SS con el
            MOV SS,AX         ; valor PILA.
```

```
            LEA SP,CAB_PILA   ; Inicializa SP con el valor CAB_PILA
                              ;
            LEA DX,CARTEL1    ; Carga DX con la direccion de comienzo
            MOV AH,9          ; Selecciona la funcion imprimir texto
            INT 21H           ; Llama a la funcion imprimir texto
                              ;
            LEA AX,TABLA      ; Carga AX con la direccion de TABLA
            PUSH AX           ; Salva la direccion en la pila
            MOV AX,N          ; Carga AX con la cant. de datos (N)
            PUSH AX           ; Salva la cant. de datos en la pila
            CALL ORDEN_DESC   ; Llama a la sub. que ordena los datos
                              ;
            LEA DX,CARTEL2    ; Carga DX con la direccion de comienzo
            MOV AH,9          ; Selecciona la funcion imprimir texto
            INT 21H           ; Llama a la funcion imprimir texto
                              ;
            MOV AH,4CH        ; Inicializa la funcion Retorno al DOS
            INT 21H           ; Llama a la funcion Retorno al DOS
                              ;
PROG  ENDS                    ; Fin del segmento PROG
END   COMENZAR                ; Direccion de transferencia luego de
                              ; cargar el programa en memoria.
```

2.3.2 Contenido del archivo ORDENDES.ASM

```
                              ; SUBRUTINA ORDEN_DESC
                              ;
                              ; Esta subrutina recibe en la pila la
                              ; direccion de una tabla de datos y
                              ; la extension de la misma. Los datos se
                              ; ordenan de mayor a menor, quedando
                              ; la tabla en su posicion original en
                              ; memoria. Los registros utilizados
                              ; por la subrutina se salvan en la pila
                              ; para luego recuperarse.
                              ;
PROG  SEGMENT  PUBLIC         ; Define un segmento llamado PROG
ASSUME CS:PROG                ; Asocia CS al segmento PROG
PUBLIC ORDEN_DESC             ; Define el rotulo ORDEN_DESC publico
                              ;
ORDEN_DESC PROC NEAR          ; Comienza proced. llamado ORDEN_DESC
            PUSH BP           ; Salva el registro BP en la pila
            MOV BP,SP         ; Carga BP con el puntero SP actual
            PUSH AX           ; Salva el registro AX en la pila
            PUSH BX           ; Salva el registro BX en la pila
            PUSH CX           ; Salva el registro CX en la pila
            PUSH SI           ; Salva el registro SI en la pila
            PUSH DI           ; Salva el registro DI en la pila
            MOV BX,[BP+6]     ; Mueve TABLA al índice BX
            MOV CX,[BP+4]     ; Mueve N al contador CX
            DEC CX            ; Decrementa el contador
LOOP1:      MOV DI,CX         ; Salva CX en DI p/ la prox. iteracion
            MOV SI,0          ; Limpia SI
LOOP2:      MOV AX,[BX][SI]   ; Mueve un dato de la tabla a AX
            CMP AX,[BX][SI+2] ; Compara con el dato sig. en la tabla
            JGE SEGUIR        ; Si el orden es correcto seguir
            XCHG AX,[BX][SI+2]; Intercambia datos p/ lograr el orden
            MOV [BX][SI],AX   ;
SEGUIR:     ADD SI,2          ; Increm. indice p/ la prox. iteracion
            LOOP LOOP2        ; Retorna a LOOP2 hasta que CX sea cero
            MOV CX,DI         ; Restaura CX
```

```
            LOOP LOOP1        ; Retorna a LOOP1 hasta CX (DI)sea cero
            POP DI            ; Restaura el registro DI
            POP SI            ; Restaura el registro SI
            POP CX            ; Restaura el registro CX
            POP BX            ; Restaura el registro BX
            POP AX            ; Restaura el registro AX
            POP BP            ; Restaura el registro BP
            RET 4             ; Retorna al programa principal e
                              ; incrementa en 4 el registro SP
                              ;
ORDEN_DESC        ENDP        ; Fin del procedimiento ORDEN_DESC
PROG              ENDS        ; Fin del segmento PROG
END                           ; Fin del modulo a ensamblar
```

2.3.3 Proceso de ensamblado y vinculación

>path = %path%;d:\Programs\MASM51 (agrega los directorios de MASM51)

```
>masm ensayo.asm,ensayo.obj,ensayo.lst,ensayo.crf
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

  46474 + 436643 Bytes symbol space free
      0 Warning Errors
      0 Severe  Errors
```

crea ensayo.obj, ensayo.lst y ensayo.crf

```
>cref ensayo.crf,ensayo.ref
Microsoft (R) Cross-Reference Utility  Version 5.10
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

12 Symbols
```

crea ensayo.ref

```
>masm ordendes.asm,ordendes.obj,ordendes.lst,ordendes.crf
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

  46274 + 436843 Bytes symbol space free
      0 Warning Errors
      0 Severe  Errors
```

crea ordendes.obj, ordendes.lst y ordendes.crf

```
>cref ordendes.crf,ordendes.ref
Microsoft (R) Cross-Reference Utility  Version 5.10
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

7 Symbols
```

crea ordendes.ref

```
>link ensayo.obj+ordendes.obj,ensayo.exe,ensayo.map,,
Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.
```

crea ensayo.exe y ensayo.map

2.3.4 Ejecución del programa en la línea de comando DOS

```
>ensayo
```

>TECNICAS DIGITALES III - LOS DATOS HAN SIDO ORDENADOS

Antes de ordenar los datos, con el programa TD se puede observar la tabla de datos, que contiene los 5 valores cargados por defecto.

```
ds:0100 05 00 01 00 02 00 03 00 ♣ ☺ ☻ ♥
ds:0108 04 00 05 00 54 45 43 4E ♦ ♣ TECN
```

La tabla de datos ordenada luego de la ejecución queda así:

```
447C:0100 05 00 05 00 04 00 03 00 ♣ ♣ ♦ ♥
447C:0108 02 00 01 00 54 45 43 4E ☻ ☺ TECN
```

Observe que para la ejecución se asignó un valor a DS.

2.3.5 Contenido del archivo ENSAYO.LST

```
Microsoft (R) Macro Assembler Version 5.10

                                ; PROGRAMA ENSAYO
                                ;
                                ; Este programa crea una tabla de
                                ; datos y llama la subrutina ORDEN_DESC
                                ; pasando la direccion del primer dato
                                ; y la cantidad de datos por la pila.
                                ;
0000                            DATOS SEGMENT
                                ; Define un segmento llamado DATOS
0000  0005                      N     DW  5
                                ; Define la variable N con el valor 5
0002  0001 0002 0003 0004       TABLA DW 1,2,3,4,5
                                ; Define la variable TABLA con el valor 1
      0005
                                ; y asigna valores a cuatro palabras mas
000C  54 45 43 4E 49 43         CARTEL1 DB 'TECNICAS DIGITALES III$'
      41 53 20 44 49 47
      49 54 41 4C 45 53
      20 49 49 49 24
0023  20 2D 20 4C 4F 53         CARTEL2 DB ' - LOS DATOS HAN SIDO ORDEN
                                ADOS$'
      20 44 41 54 4F 53
      20 48 41 4E 20 53
      49 44 4F 20 4F 52
      44 45 4E 41 44 4F
      53 24
0043                            DATOS ENDS
                                ; Fin del segmento DATOS
                                ;
0000                            PILA SEGMENT STACK
                                ; Define un segmento PILA de tipo STACK
0000  0080[                           DW  128 DUP (?)
                                ; Reserva 128 palabras
      ????
                ]

0100                            CAB_PILA LABEL WORD
```

```
                                ; Nombra CAB_PILA al tope del stack
31 0100                         PILA ENDS
                                ; Fin del segmento PILA
32                              ;
33 0000                         PROG SEGMENT PUBLIC
                                ; Define un segmento PROG de tipo PUBLIC
34                              EXTRN ORDEN_DESC:NEAR
                                ; Define subrutina externa y proxima
35                              ASSUME CS:PROG,DS:DATOS,SS:PILA
                                ; Asocia CS al segmento PROG ,DS al seg.
36                              ; DATOS y SS al seg. PILA
37                              ;
38 0000  B8 ---- R              COMENZAR: MOV AX,DATOS
                                ; Inicializa el registro DS con el valor
                                ; DATOS, obtenido por el ensamblador
39 0003  8E D8                           MOV DS,AX
40 0005  B8 ---- R                       MOV AX,PILA
                                ; Inicializa el registro SS con el valor
                                ; PILA , obtenido por el ensamblador
41 0008  8E D0                           MOV SS,AX
42 000A  8D 26 0100 R                    LEA SP,CAB_PILA
                                ; Inicializa SP con el valor CAB_PILA
43                              ;
44 000E  8D 16 000C R                    LEA DX,CARTEL1
                                ; Carga DX con la direccion de comienzo
45 0012  B4 09                           MOV AH,9
                                ; Selecciona la funcion imprimir texto
46 0014  CD 21                           INT 21H
                                ; Llama a la funcion imprimir texto
47                              ;
48 0016  8D 06 0002 R                    LEA AX,TABLA
                                ; Carga AX con la direccion de TABLA
49 001A  50                              PUSH AX
                                ; Salva la direccion en la pila
50 001B  A1 0000 R                       MOV AX,N
                                ; Carga AX con la cantidad de datos (N)
51 001E  50                              PUSH AX
                                ; Salva la cantidad de datos en la pila
52 001F  E8 0000 E                       CALL ORDEN_DESC
                                ; Llama a la subr. que ordena los datos
53                              ;
54 0022  8D 16 0023 R                    LEA DX,CARTEL2
                                ; Carga DX con la direccion de comienzo
55 0026  B4 09                           MOV AH,9
                                ; Selecciona la funcion imprimir texto
56 0028  CD 21                           INT 21H
                                ; Llama a la funcion imprimir texto
57                              ;
58 002A  B4 4C                           MOV AH,4CH
                                ; Inicializa la funcion Retorno al DOS
59 002C  CD 21                         INT 21H
                                ; Llama a la funcion Retorno al DOS
60                              ;
61 002E                         PROG ENDS
                                ; Fin del segmento PROG
62                              END COMENZAR
                                ; Direccion de transf. luego de cargar

Symbols-1
Segments and Groups:
```

```
               N a m e            Length      Align      Combine Class

DATOS  . . . . . . . . . . . . .   0043  PARA  NONE
PILA . . . . . . . . . . . . . .   0100  PARA  STACK
PROG . . . . . . . . . . . . . .   002E  PARA  PUBLIC

Symbols:
               N a m e            Type   Value       Attr

CAB_PILA . . . . . . . . . . . .   L WORD      0100  PILA
CARTEL1  . . . . . . . . . . . .   L BYTE      000C  DATOS
CARTEL2  . . . . . . . . . . . .   L BYTE      0023  DATOS
COMENZAR . . . . . . . . . . . .   L NEAR      0000  PROG

N  . . . . . . . . . . . . . . .   L WORD      0000  DATOS

ORDEN_DESC . . . . . . . . . . .   L NEAR      0000  PROG  External

TABLA  . . . . . . . . . . . . .   L WORD      0002  DATOS

@CPU . . . . . . . . . . . . . .   TEXT  0101h
@FILENAME  . . . . . . . . . . .   TEXT  ensayo
@VERSION . . . . . . . . . . . .   TEXT  510

     50 Source  Lines
     50 Total   Lines
     15 Symbols

  46474 + 436643 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

2.3.6 Contenido del archivo ENSAYO.REF

```
Microsoft Cross-Reference  Version 5.10
Symbol Cross-Reference      (# definition, + modification)  Cref-1

@CPU . . . . . . . . . . . . . .    1#
@VERSION . . . . . . . . . . . .    1#

CAB_PILA . . . . . . . . . . . .   30#    42
CARTEL1. . . . . . . . . . . . .   13#    44
CARTEL2. . . . . . . . . . . . .   17#    54
COMENZAR . . . . . . . . . . . .   38#    62

DATOS. . . . . . . . . . . . . .    8#    23     35     38

N. . . . . . . . . . . . . . . .    9#    50

ORDEN_DESC . . . . . . . . . . .   34#    52

PILA . . . . . . . . . . . . . .   25#    31     35     40
PROG . . . . . . . . . . . . . .   33#    35     61

TABLA. . . . . . . . . . . . . .   10#    48


 12 Symbols
```

2.3.7 Contenido del archivo ORDENDES.LST

```
Microsoft (R) Macro Assembler Version 5.10

1                                   ; SUBRUTINA ORDEN_DESC
2                                   ;
3                                   ; Esta subrutina recibe en la pila la
4                                   ; direccion de una tabla de datos y
5                                   ; la extension de la misma. Los datos se
6                                   ; ordenan de mayor a menor, quedando
7                                   ; la tabla en su posicion original en
8                                   ; memoria. Los registros utilizados
9                                   ; por la subrutina se salvan en la pila
10                                  ; para luego recuperarse.
11                                  ;
12 0000                             PROG  SEGMENT  PUBLIC
                                    ; Define un segmento llamado PROG
13                                  ASSUME CS:PROG
                                    ; Asocia CS al segmento PROG
14                                  PUBLIC ORDEN_DESC
                                    ; Define rotulo ORDEN_DESC publico
15                                  ;
16 0000                             ORDEN_DESC  PROC  NEAR
                                    ; Comienza el procedimiento ORDEN_DESC
17 0000  55                               PUSH BP
                                    ; Salva el registro BP en la pila
18 0001  8B EC                           MOV BP,SP
                                    ; Carga indice BP con puntero SP actual
19 0003  50                               PUSH AX
                                    ; Salva el registro AX en la pila
20 0004  53                               PUSH BX
                                    ; Salva el registro BX en la pila
21 0005  51                               PUSH CX
                                    ; Salva el registro CX en la pila
22 0006  56                               PUSH SI
                                    ; Salva el registro SI en la pila
23 0007  57                               PUSH DI
                                    ; Salva el registro DI en la pila
24 0008  8B 5E 06                        MOV BX,[BP+6]
                                    ; Mueve TABLA al indice BX
25 000B  8B 4E 04                        MOV CX,[BP+4]
                                    ; Mueve N al contador CX
26 000E  49                               DEC CX
                                    ; Decrementa el contador
27 000F  8B F9                      LOOP1:MOV DI,CX
                                    ; Salva CX en DI para proxima iteracion
28 0011  BE 0000                         MOV SI,0
                                    ; Limpia SI
29 0014  8B 00                      LOOP2:MOV AX,[BX][SI]
                                    ; Mueve un dato de la tabla a registro AX
30 0016  3B 40 02                        CMP AX,[BX][SI+2]
                                    ; Compara con dato siguiente en la tabla
31 0019  7D 05                           JGE SEGUIR
                                    ; Si el orden es correcto seguir
32 001B  87 40 02                        XCHG AX,[BX][SI+2]
                                    ; Intercambia datos para lograr el orden
33 001E  89 00                           MOV [BX][SI],AX
                                    ;
34 0020  83 C6 02                   SEGUIR:ADD SI,2
                                    ; Increm. indice para proxima iteracion
35 0023  E2 EF                           LOOP LOOP2
```

```
                                ; Retorna a LOOP2 hasta que CX sea cero
36 0025  8B CF                        MOV CX,DI
                                ; Restaura CX
37 0027  E2 E6                        LOOP LOOP1
                                ; Retorna a LOOP1 hasta CX (DI) sea cero
38 0029  5F                           POP DI
                                ; Restaura el registro DI
39 002A  5E                           POP SI
                                ; Restaura el registro SI
40 002B  59                           POP CX
                                ; Restaura el registro CX
41 002C  5B                           POP BX
                                ; Restaura el registro BX
42 002D  58                           POP AX
                                ; Restaura el registro AX
43 002E  5D                           POP BP
                                ; Restaura el registro BP
44 002F  C2 0004                      RET 4
                                ; Retorna al prog. principal e incrementa
45                              ; en 4 el registro SP
46                              ;
47 0032                         ORDEN_DESC  ENDP
                                ; Fin del procedimiento ORDEN_DESC
48 0032                         PROG        ENDS
                                ; Fin del segmento PROG
49                                    END
                                ; Fin del modulo a ensamblar

Symbols-1
Segments and Groups:

                N a m e             Length       Align      Combine Class

PROG . . . . . . . . . . . . . .    0032  PARA  PUBLIC

Symbols:
                N a m e             Type   Value       Attr

LOOP1  . . . . . . . . . . . . .    L NEAR      000F  PROG
LOOP2  . . . . . . . . . . . . .    L NEAR      0014  PROG

ORDEN_DESC . . . . . . . . . . .    N PROC      0000  PROG  Global      Length
= 0032

SEGUIR . . . . . . . . . . . . .    L NEAR      0020  PROG

@CPU . . . . . . . . . . . . . .    TEXT  0101h
@FILENAME  . . . . . . . . . . .    TEXT  ordendes
@VERSION . . . . . . . . . . . .    TEXT  510

     49 Source  Lines
     49 Total   Lines
     10 Symbols

  46274 + 436843 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

2.3.8 Contenido del archivo ORDENDES.REF

```
Microsoft Cross-Reference  Version 5.10

Symbol Cross-Reference      (# definition, + modification)  Cref-1

@CPU . . . . . . . . . . . . . . .    1#
@VERSION . . . . . . . . . . . . .    1#

LOOP1. . . . . . . . . . . . . . .   27#    37
LOOP2. . . . . . . . . . . . . . .   29#    35

ORDEN_DESC . . . . . . . . . . . .   14     16#    47

PROG . . . . . . . . . . . . . . .   12#    13     48

SEGUIR . . . . . . . . . . . . . .   31     34#
```

2.3.9 Contenido del archivo ENSAYO.MAP

```
 Start  Stop   Length Name                   Class
 00000H 00042H 00043H DATOS
 00050H 0014FH 00100H PILA
 00150H 001B1H 00062H PROG

Program entry point at 0015:0000
```

2.4 EJEMPLOS DE PROGRAMACION EN C Y ASSEMBLY

A continuación se muestra un ejemplo de programa en C y una subrutina en assembly, que luego serán compilados, ensamblados y vinculados en la PC.

Los archivos que se mencionan están incluidos en el CD de Técnicas Digitales III y en la página web de la materia (http://www.frsn.utn.edu.ar/tecnicas3).

2.4.1 Contenido del archivo PRIMOS.C

```
/* devuelve los numeros primos hasta el 1000 */
#include <stdio.h>
extern int primo(int n);  /* definimos el prototipo de la funcion */
main()
{
  int i,n;
  n=1000;
  for (i=2;i<=n;i++)
    if (primo(i))         /* llamamos a la funcion externa en asm */
      printf("%d ",i);
  return 0;
}
```

2.4.2 Contenido del archivo CALCULA.ASM

```
; Decide si un numero que se le pasa como parametro es primo o no
; Devuelve false (0) o true (!0) al programa en C
; Se invoca desde C: primo(i) siendo i de tipo int (16 bits)
;
.MODEL SMALL                 ;Usamos el modelo SMALL y debe coincidir
                             ;con el usado en C
.CODE                        ;Solo usamos el segmento de codigo
```

```
PUBLIC      _primo          ;Se declara el procedimiento como publico
                            ;para que pueda ser llamado desde otro modulo
_primo      PROC NEAR       ;
            PUSH BP         ;Guardamos BP
            MOV BP, SP      ;Recuperamos en BP el puntero de la pila
            MOV SI, [BP+4]  ;Recuperamos en SI el parámetro que le hemos
                            ;pasado [BP+4] (2 del BP + 2 de la direccion
                            ;de retorno)
            MOV BX, 2       ;BX contiene el divisor (de 2 a SI)
BUCLE:      XOR DX, DX      ;Dividendo DX&AX
            MOV AX, SI      ;Divisor BX
            CMP AX, BX      ;Si ya se ha dividido por todos
            JE  PRIMO       ;indica que es primo
            DIV BX          ;Dividimos DX&AX / BX
            OR DX, DX       ;DX contiene el resto de la division
            JZ NOPRIMO      ;si es 0 el numero no es primo
            INC BX          ;Incrementamos el divisor
            JMP BUCLE       ;volvemos a dividir
PRIMO:      MOV AX, 1       ;Colocamos el valor de retorno a 1 (true en C)
            JMP FIN         ;
NOPRIMO:    XOR AX, AX      ;Colocamos el valor de retorno a 0 (falso en C)
FIN:        POP BP          ;Quitamos el valor de BP que guardamos
                            ;al principio
            RET             ;Retornamos
_primo      ENDP            ;
            END             ;
```

2.4.3 Proceso de ensamblado, compilación y vinculación

>path = %path%;d:\Programs\TC201 (agrega los directorios de TC201)

>tasm /ml calcula.asm

```
Turbo Assembler  Version 1.0  Copyright (c) 1988 by Borland International

Assembling file:   CALCULA.ASM
Error messages:    None
Warning messages:  None
Remaining memory:  483k
```

crea calcula.obj

>tcc –Eprimos.exe primos.c calcula.obj

```
Turbo C  Version 2.01  Copyright (c) 1987, 1988 Borland International
primos.c:
```

crea primos.obj

```
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
         Available memory 424408
```

crea primos.exe

2.4.4 Ejecución del programa en la línea de comando DOS

>**primos**

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101
103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197
```

```
199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311
313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431
433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541 547 557
563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661
673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809
811 821 823 827 829 839 853 857 859 863 877 881 883 887 907 911 919 929 937
941 947 953 967 971 977 983 991 997
```

CAPÍTULO 3

MEDIOS DE ALMACENAMIENTO DE DATOS

3.1 CLASIFICACION Y APLICACIONES

3.1.1 Clasificación

Los medios masivos de almacenamiento de datos se caracterizan por dos importantes propiedades: cantidad de información y transportabilidad. Las aplicaciones que aprovechan la posibilidad de almacenamiento de gran cantidad de información son, entre otras, el sistema operativo, las bases de datos, los archivos temporarios, las aplicaciones multimedia (imagen y sonido), etc. Los medios de almacenamiento masivo lo logran a un bajo costo por unidad de información, en un espacio reducido. La transportabilidad permite la distribución y actualización de software, las copias de seguridad, los archivos permanentes, etc.

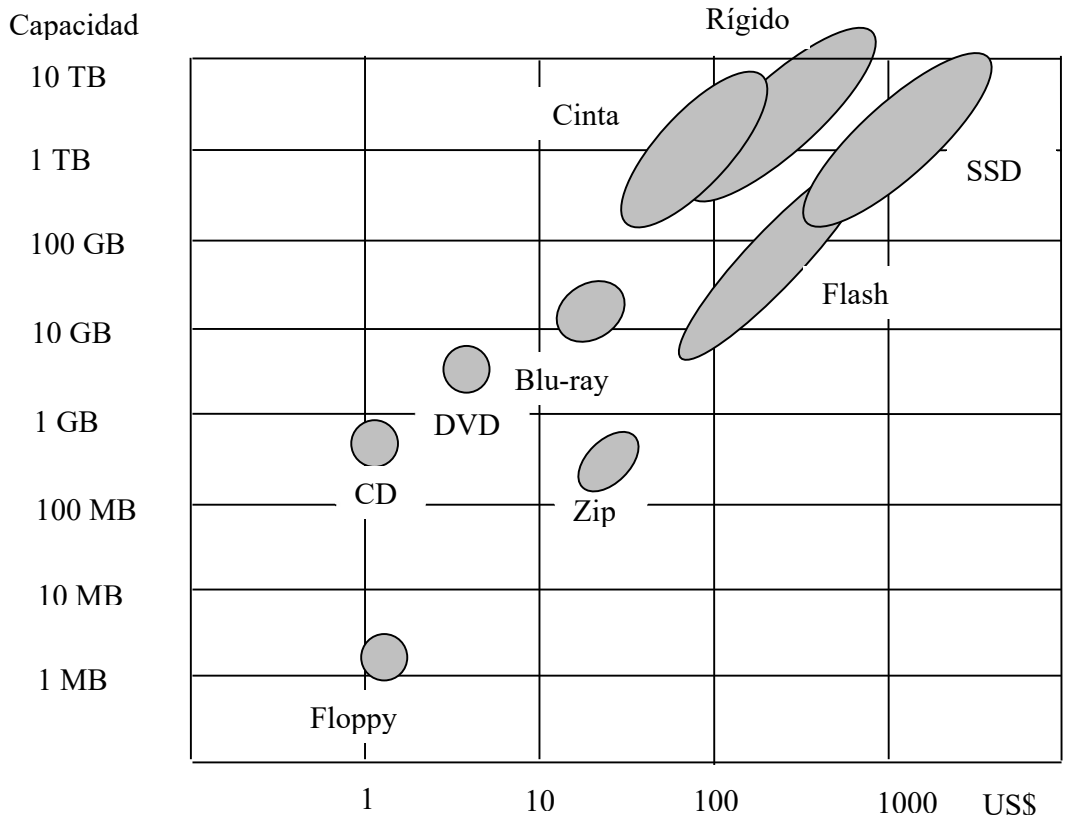


Figura 3.1.1 - Relación Costo/Capacidad

Los medios de almacenamiento masivo en uso en la actualidad son:

- discos rígidos
- cintas
- discos ópticos (CD, DVD, BLU-RAY)
- memorias flash (USB/SD)
- discos de estado sólido (SSD)

3.1.2 Aplicaciones

Cada uno de ellos tiene una aplicación específica que hace que estas distintas tecnologías puedan coexistir en el mercado.

Los discos rígidos se utilizan principalmente como *disco del sistema* cuando se emplean sistemas operativos en disco (ej. DOS, RSX, VMS, UNIX, NT, etc.). También para tener acceso rápido a bases de datos, como área de almacenamiento temporario en procesos de compilación y para alojar tareas inactivas en ambientes multitarea (checkpointing).

Los menores costos de las unidades SSD están cambiando el estándar para soporte del sistema operativo. Dada su mayor velocidad, comparada con la de los discos rígidos, están siendo empleados por la mayoría de los usuarios corporativos, nuevas laptops, gamers e incluso usuarios particulares de desktops.

Las cintas se usan para respaldar discos rígidos, como medio de almacenamiento de archivos históricos o bases de datos secuenciales muy grandes y como medio de transporte y distribución de software.

Los discos flexibles se usaban en sistemas de bajo costo como transporte y distribución de software y como respaldo de archivos de tamaño reducido. Actualmente en desuso.

Los discos Zip, una mezcla de disco flexible con tecnología de disco rígido, llegaron a tener capacidades de 250 MB en un medio removible. Actualmente en desuso.

Los discos ópticos (CD-ROM) se utilizan (cada vez menos) como medio de distribución de música y drivers. Existen versiones CD-RW.

Los DVD y Blu-Ray compiten con las cintas en su capacidad de almacenamiento y se usan (cada vez menos) como backup de datos, transporte de SW y distribución de multimedia.

3.2 INTERFAZ CON EL PROCESADOR

3.2.1 Esquema general

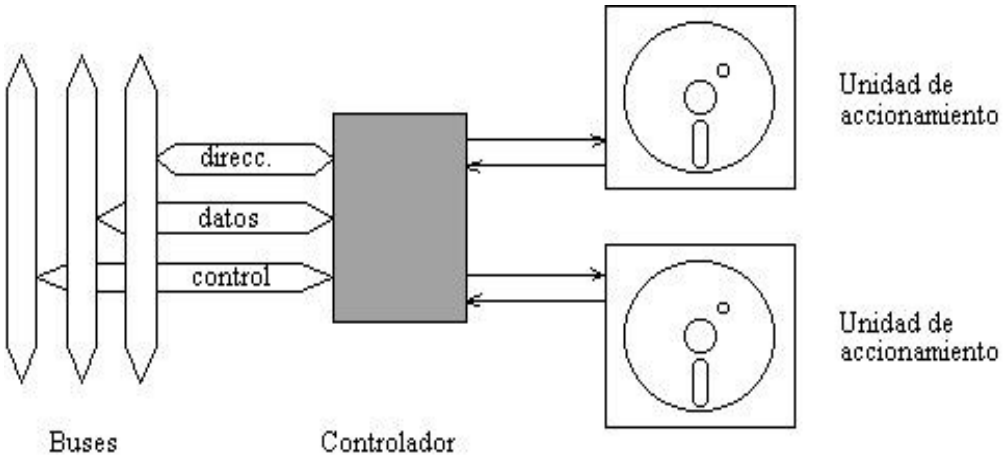


Figura 3.2.1 - Controlador de periféricos - Diagrama en bloques

En este esquema, el controlador puede manejar más de una unidad de accionamiento. Existen también las unidades con controlador incorporado, todos unidos a un bus, eliminando el cuello de botella del controlador cuando hay unidades múltiples.

El controlador contiene la lógica de DMA, memoria incorporada (caché) para compensar las diferencias de velocidad entre el medio magnético y el bus, lógica de interrupciones, conversión serie/paralelo, manejo de errores, etc. Actualmente están controlados por un microprocesador de alta velocidad o circuitos dedicados y realizan funciones de optimización.

3.2.2 Tipos de controladores/interfases

IDE: Integrated Drive Electronics.

Es un controlador cuya característica principal es que su lógica se encuentra integrada en la unidad de accionamiento. Una ventaja de este método radica en que los errores se detectan y corrigen en forma local, sin afectar la comunicación con el procesador. Su performance se sitúa entre los 4 y los 8 MB/s, cuando se encuentra conectado al bus ISA. Una limitación es que se pueden conectar sólo dos discos IDE.

La versión mejorada, Enhanced IDE, soporta hasta 4 periféricos, dos en el controlador primario y dos en el secundario. En cada controlador se debe seleccionar un dispositivo como "master" y el otro como "slave", usualmente mediante un puente en el dispositivo.

Figura 3.2.2 – Conector de la interfaz IDE

UATA/UDMA:

En las placas madre que lo permitían (a partir de 1999) se accedía a los discos rígidos por acceso directo a memoria, a la velocidad de 33 MB/s (UDMA 33), 66 MB/s, 100 MB/s y 133 MB/s (UDMA 133) con lo que se alcanzó el límite del bus PCI (133 MHz). La conexión a partir de ATA66 requiere un cable plano especial de 80 conductores que posee los mismos conectores de 40 contactos y los otros están a masa para apantallar las señales y reducir la interferencia a esta elevada frecuencia.

SCSI: Small Computer System Interface.

En este caso existe un controlador en el bus del sistema, al que pueden conectarse hasta siete dispositivos, que pueden ser discos, cintas, CD-ROMs, aunque sólo uno de ellos se puede comunicar a la vez con el procesador. El bus de datos es de 8 bits, alcanzando velocidades de 5 MB/s. Se conecta sobre el bus ISA. La versión mejorada, SCSI-2 aumenta el ancho del bus de datos a 16 o 32 bits, pudiendo lograrse velocidades de hasta 40 MB/s.

Figura 3.2.3 – Conector de la interfaz SCSI

SAS: Serial Attached SCSI.

Es una interfaz de transferencia de datos en serie, sucesor del Small Computer System Interface (SCSI) paralelo, aunque sigue utilizando comandos SCSI para interactuar con los dispositivos SAS. Aumenta la velocidad y permite la conexión y desconexión de forma rápida. El último estándar vigente, desde 2017, es SAS-4 que opera a 22,5 Gb/s.

Figura 3.2.4 – Conector de la interfaz SAS

SATA:

Desde 2004 se utiliza la conexión Serial ATA (SATA). Se trata de una comunicación serie que opera a 1,5 Gbps, permitiendo velocidades de transferencia pico de 150 MB/s. El conector de datos es más pequeño, ya que sólo posee siete conductores, dos para transmisión, dos para recepción y tres GND y puede tener una longitud de varios metros. SATA II, una actualización de este estándar, opera a 3 Gbps (300 MB/s). En 2010 comienza a difundirse SATA III, que posee una especificación de 6 Gbps (600 MB/s). Las alimentaciones de 5 V y 12 V también poseen un conector diferente al de los antiguos IDE.

Los discos rígidos mecánicos rara vez alcanzan los 150 MB/s de velocidad de transferencia, por lo que los nuevos estándares de SATA no representan un beneficio en la performance. Sin embargo, los dispositivos de estado sólido (SSD) si aprovechan estos incrementos de velocidad de transferencia.

Figura 3.2.5 – Conector de la interfaz SATA

La especificación AHCI define a los fabricantes de hardware una estructura de memoria para intercambiar datos entre la memoria del sistema host y los dispositivos de almacenamiento conectados. AHCI proporciona a los desarrolladores de software y diseñadores de hardware un método estándar para detectar, configurar y programar adaptadores SATA/AHCI. AHCI es independiente del estándar SATA 3Gb/s, si bien expone las capacidades avanzadas de SATA (como NCQ y Hot Swap) de tal manera que los sistemas puedan utilizarlos.

NCQ es una forma de recuperar los datos según la facilidad de obtenerlos de un disco en rotación, como se ve en la figura, reduciendo los tiempos totales, aunque no se respete el orden

original de las solicitudes. Se entenderá mejor cuando veamos el funcionamiento de una unidad de discos rígidos.

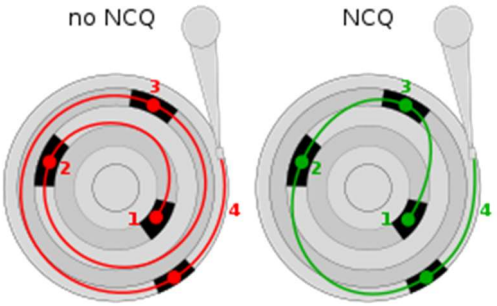


Figura 3.2.6 – Comparativa sin NCQ vs con NCQ

Hot swap está permitido por diseño en SATA. Esta interfaz soporta la conexión y desconexión en uso, es decir con la alimentación aplicada. Esto es muy útil en sistemas redundantes, como se explicará en el párrafo 3.9, al final de este capítulo.

M.2:

M.2 anteriormente conocido como NGFF (por las siglas en inglés Next Generation Form Factor o factor de forma de nueva generación) es una especificación para tarjetas de expansión interna de computadoras. Las especificaciones de M.2 son flexibles, lo que permite diferentes longitudes y anchos de módulos para las unidades de estado sólido en general y para su uso en dispositivos más pequeños como ultrabooks o tablets.

La interfaz del bus de datos que está detrás del conector M.2 es una PCI Express 3.0 (hasta cuatro slots PCI Express), Serial ATA 3.0 y USB 3.0 (un puerto lógico individual por cada uno de los dos últimos). Está a merced del fabricante del dispositivo o del puerto M.2 elegir qué interfaces se soportarán, dependiendo del nivel deseado tanto del dispositivo como del receptor. El conector M.2 puede presentar distintas muescas de codificación que denotan tanto diferentes capacidades como propósitos, evitando así el uso de módulos M.2 en dispositivos incompatibles.

Para dispositivos SSD que usen M.2 a través de PCI Express 3.0 4x, la máxima velocidad teórica es de 3938 MB/s. Si, en cambio, el puerto soporta PCI Express 4.0, la velocidad asciende a 7877 MB/s. En 2021 los SSD comerciales más rápidos pueden operar cerca de 7000 MB/s.

Figura 3.2.7 – Conector de la interfaz M.2

A continuación, una comparativa de las distintas interfases (de Wikipedia, editado).

Name	Raw data rate	Data rate	Max. cable length	Power provided	Devices per channel
eSATA	6 Gbit/s	600 MB/s	2 m	No	1 (15 with a port multiplier)
eSATAp	6 Gbit/s	600 MB/s	1 m	5 V, 12 V	
SATA Express	16 Gbit/s	1.97 GB/s	1 m	No	
SATA revision 3.0	6 Gbit/s	600 MB/s			
SATA revision 2.0	3 Gbit/s	300 MB/s			
SATA revision 1.0	1.5 Gbit/s	150 MB/s			1
PATA (IDE) 133	1.064 Gbit/s	133.3 MB/s	0.46 m	5 V (only 2.5-inch drive 44-pin conn)	2
SAS-4	22.5 Gbit/s	2.25 GB/s	10 m	Backplane conn only	1 (> 65k with expanders)
SAS-3	12 Gbit/s	1.2 GB/s			
SAS-2	6 Gbit/s	600 MB/s			
SAS-1	3 Gbit/s	300 MB/s			
IEEE 1394 (FireWire) 3200	3.144 Gbit/s	393 MB/s	100 m	15 W, 12–25 V	63 (with a hub)
IEEE 1394 (FireWire) 800	786 Mbit/s	98.25 MB/s	100 m		
IEEE 1394 (FireWire) 400	393 Mbit/s	49.13 MB/s	4.5 m		
USB 3.2 (Gen 2x2)	20 Gbit/s	2.44 GB/	1 m	Yes 100 W, 5, 12 or 20 V	127 (with a hub)
USB 3.1 (Gen 2)	10 Gbit/s	1.22 GB/s	1 m	100 W, 5, 12 or 20 V	127 (with a hub)
USB 3.0 (USB 3.2, Gen 1)	5 Gbit/s	610 MB/s	2 m	4.5 W, 5 V	
USB 2.0	480 Mbit/s	58 MB/s	5 m	2.5 W, 5 V	
USB 1.1	12 Mbit/s	1.5 MB/s	3 m	Yes	
SCSI Ultra-320	2.56 Gbit/s	320 MB/s	12 m	Only with SCA Backplane	15 excl. host bus adapter/host
PCI Express 1.0 (×4 link)	10 Gbit/s	1 GB/s	n/a	yes	n/a
PCI Express 2.0 (×4 link)	20 Gbit/s	2 GB/s	n/a	yes	n/a
PCI Express 3.0 (×4 link)	32 Gbit/s	3.938 GB/s	n/a	yes	n/a
PCI Express 4.0 (×4 link)	64 Gbit/s	7.8775 GB/s	n/a	yes	n/a

Figura 3.2.8 – Evolución de los estándares

3.3 METODOS DE GRABACION MAGNETICA

3.3.1 Introducción

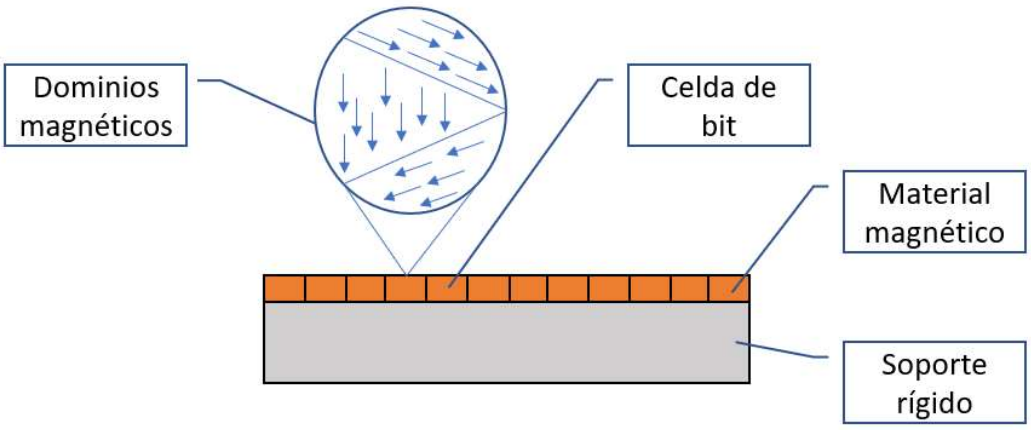


Figura 3.3.1 – Medios de almacenamiento magnéticos

El material donde se almacenan los datos es una fina capa de Níquel, un material ferromagnético con alta histéresis. Los minúsculos dominios magnéticos son grupos de dipolos que se orientan en función del campo aplicado y luego de retirado el mismo se mantienen con la misma orientación.

Las celdas de bit son las mínimas áreas que pueden orientarse sin influir en las vecinas.

El material magnético yace sobre una superficie sólida, en general de aluminio, que da el nombre a uno de los medios conocidos: el disco rígido.

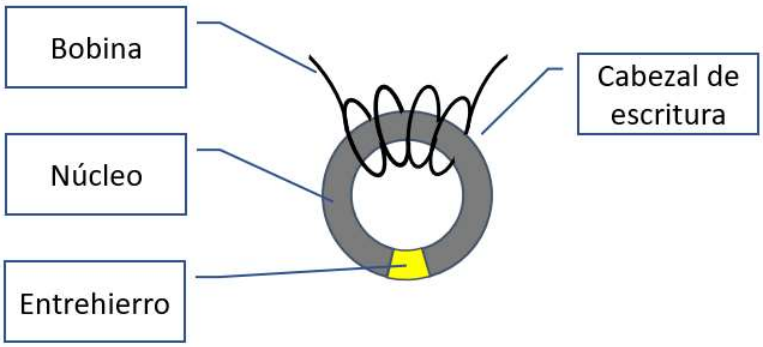


Figura 3.3.2 – Cabezal de grabación magnética

El cabezal de escritura consiste en un núcleo de material ferromagnético de baja histéresis sobre el cual esté arrollado un bobinado. Cuando se aplica corriente eléctrica al bobinado, el núcleo se magnetiza como un electroimán. Cuando se retira la corriente, el mismo se desmagnetiza.

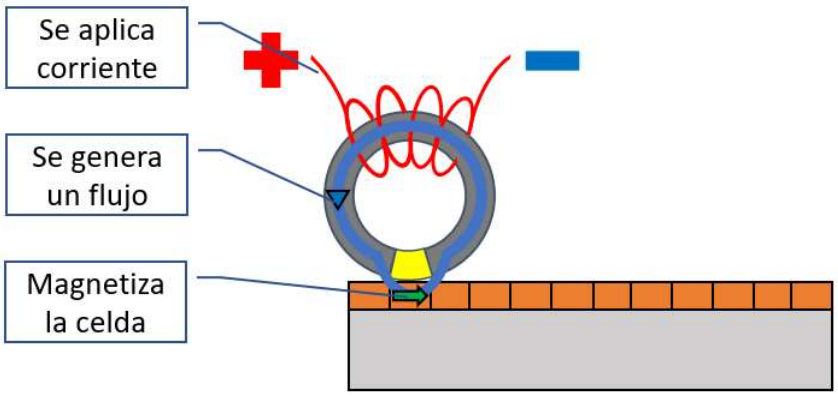


Figura 3.3.3 – Método de grabación

Como se ve en la figura, las líneas de flujo magnético se desvían del entrehierro para magnetizar la superficie de grabación. Este efecto se ve favorecido por la utilización de un material diamagnético (shim) en el entrehierro, el que, al llenar ese espacio, impide la acumulación de partículas extrañas.

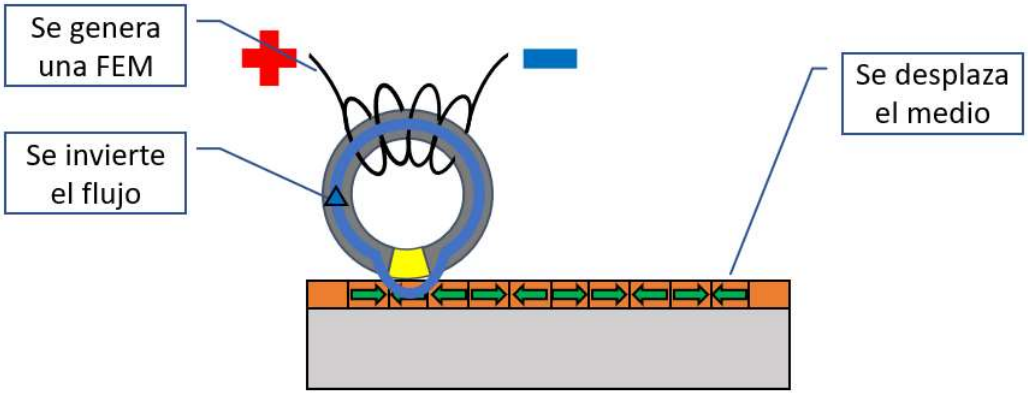


Figura 3.3.4 – Método de lectura

El mismo cabezal se puede utilizar para la lectura. Como se ve en la figura, cuando el medio se mueve debajo del entrehierro, las distintas direcciones del campo magnético grabado generan variaciones de flujo en el núcleo. A su vez, estas variaciones de flujo generan FEM en la bobina. Esta FEM se utiliza, luego de ser procesada y amplificada, para reconstruir la información previamente grabada en el medio.

En los discos flexibles y cintas, la cabeza roza contra la superficie del medio magnético; en los discos rígidos, la misma está separada una pequeña distancia, llamada “altura de vuelo”, explicada en detalle en 3.5.

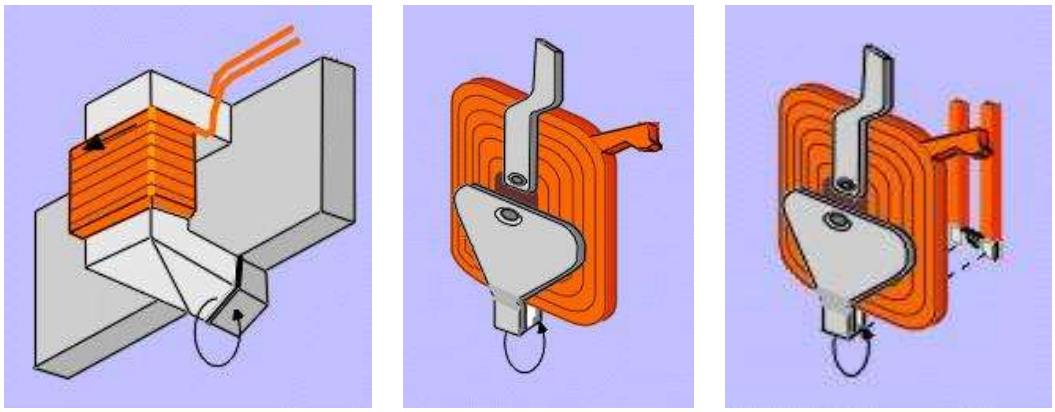


Figura 3.3.5 – Cabezales – A: Inductivo – B: Thin film – C: MR/GMR/TMR

En la actualidad, los discos rígidos poseen un cabezal de lectura separado del de escritura. El mismo fue evolucionando desde el efecto Hall, el MR (magneto-resistivo), pasando por el GMR (magneto-resistivo-gigante) y finalmente el TMR (magneto-resistivo-túnel).

La variación de tensión que se obtiene en estos sensores es mucho mayor que la fem en una bobina, para una transición de flujo dada, lo que permite detectar los menores efectos magnéticos que se dan al reducir las celdas y aumentar la densidad de grabación.

Efecto Hall

Cuando por un material circula una corriente eléctrica y estando este mismo material en el seno de un campo magnético, se comprueba que aparece una fuerza en los portadores de carga que los reagrupa dentro del material; esto es, los portadores de carga se desvían y agrupan a un lado del material conductor o semiconductor, apareciendo así una variación de resistividad.

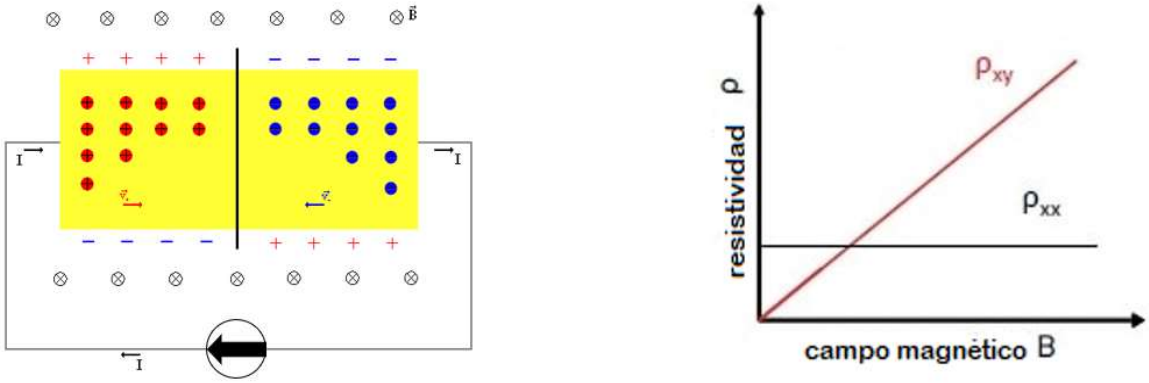


Figura 3.3.6 – Efecto Hall

El efecto magnetorresistivo se basa en la propiedad de un dispositivo susceptor magnético de película delgada hecho de un material de aleación ferromagnética de variar su resistencia en la presencia de un campo magnético o magnetorresistencia.

Este sensor tiene un coeficiente de temperatura pequeño y un rendimiento estable lo que lo hace apropiado para los discos rígidos. A diferencia del sensor de semiconductor, es más sensible a la dirección del campo magnético.

Grabación perpendicular

Desde de 2008 ya se utiliza el método de magnetización perpendicular, que logra aproximadamente el doble de densidad de grabación.

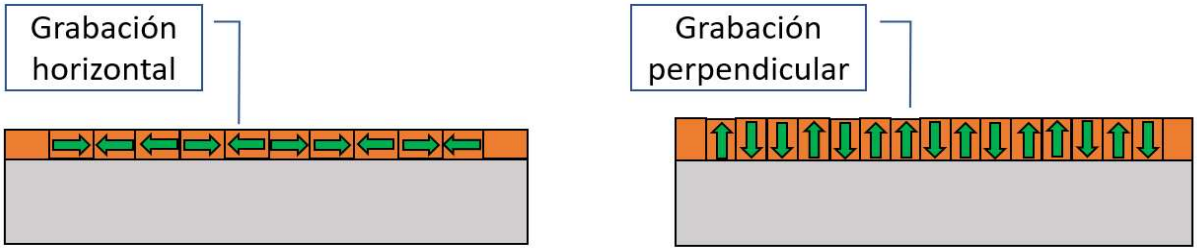


Figura 3.3.7 – Grabación horizontal vs grabación perpendicular

En la figura se puede ver la disposición del cabezal de grabación.

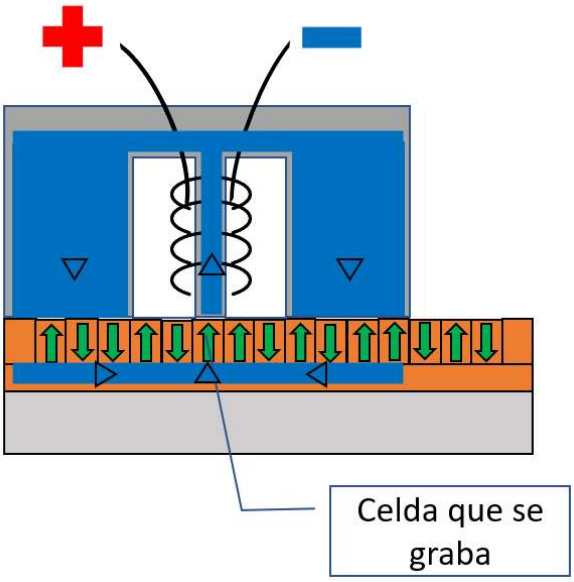


Figura 3.3.8 – Grabación perpendicular

Sólo una celda se graba en forma perpendicular. Las celdas por las que retornan las líneas de flujo no se ven afectadas porque al repartirse entre una gran superficie el flujo, el mismo es muchísimo menor al que se usó para grabarlas.

Codificación de los datos

Para codificar los datos binarios en forma de cambios o transiciones de campo magnético, se idearon distintos métodos. Comentaremos los tres principales.

3.3.2 Método FM (modulación de frecuencia)

- Transición (T) en todos los bordes de celda
- Para grabar "1": transición (T) en el centro de la celda
- Para grabar "0": sin transición (N) en el centro de la celda

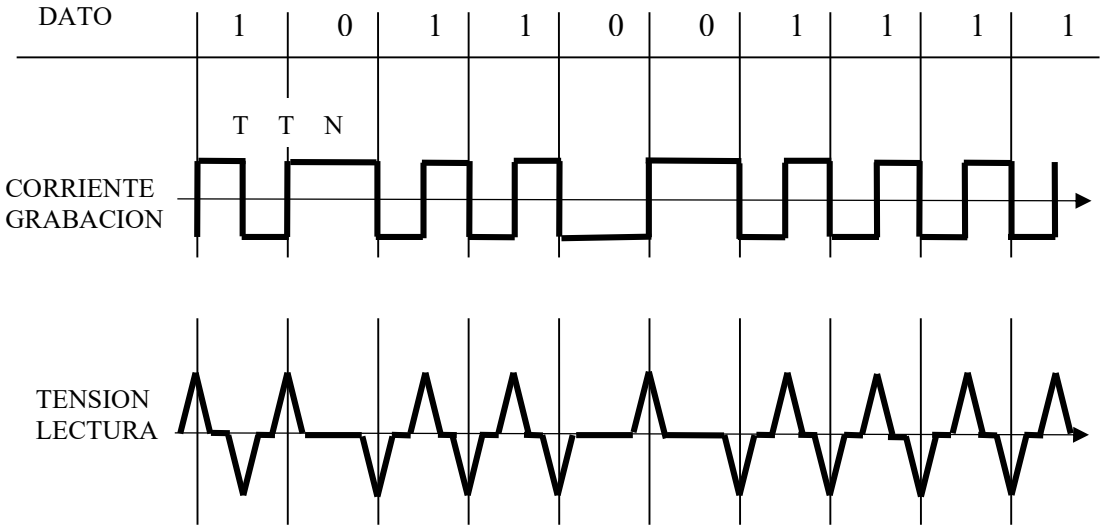


Figura 3.3.9 – Método de grabación FM

Vemos que al grabar varios "1" se obtiene una "frecuencia" doble que al grabar varios "0".

3.3.3 Método MFM (FM modificado o Miller)

- Para grabar "1": transición (T) en el centro de la celda
- Para grabar "0": transición (T) en el borde de la celda, sólo si el dato anterior era "0".

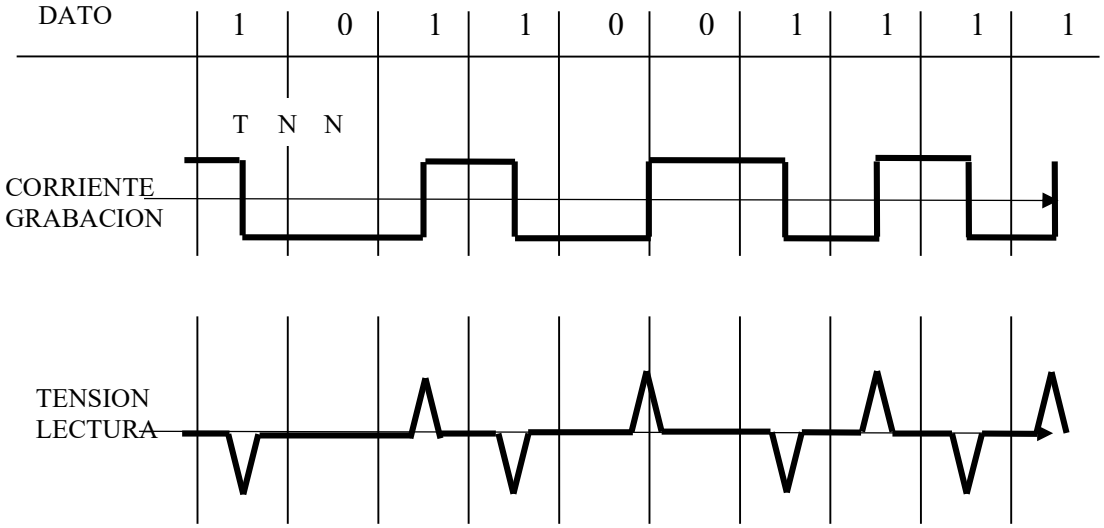


Figura 3.3.10 – Método de grabación MFM

Comparando ambos métodos, vemos que en MFM ningún dato requiere más de una transición por celda. Por el contrario, en FM se necesitan dos transiciones para grabar un "1". Teniendo en cuenta que, para un mismo medio magnético, la cantidad de información que se puede almacenar depende de la densidad de grabación (transiciones por unidad de longitud) y del método de grabación (bits por transición), con MFM se puede guardar el doble de información en la misma longitud que la que se almacenaría con FM. Ambos métodos permiten recuperar la señal de reloj, que determina el momento oportuno para leer la información de la celda.

3.3.4 Método RLL (Run Lenght Limited)

Aprovechando implementaciones de hardware más complejas, se pueden utilizar técnicas más elaboradas como RLL (Run Lenght Limited) y ARLL (Advanced RLL).

En RLL 2,7 (inventado por IBM en 1979) la codificación se basa en una tabla que traduce las distintas combinaciones de unos y ceros que se pretende grabar a una secuencia especial de transiciones de flujo. Con esto se logra mejorar la relación entre información y señal de reloj, aumentando en un 50 % la densidad de grabación respecto de MFM.

RLL n,k significa que la mínima cantidad de N seguidos entre dos T será de n y la máxima será k. La mínima determina la densidad de grabación y la máxima, la facilidad de recuperar el reloj. En la figura, que corresponde a Seagate ST11R IBM, se resaltan n=2 y k=7 para RLL 2,7.

Conjuntos de datos	Transiciones de flujo
10	NTNN
11	TNNN
000	NNNTNN
010	TNNTNN
011	NNTNNN
0010	NNTNNTNN
0011	NNNNTNNN

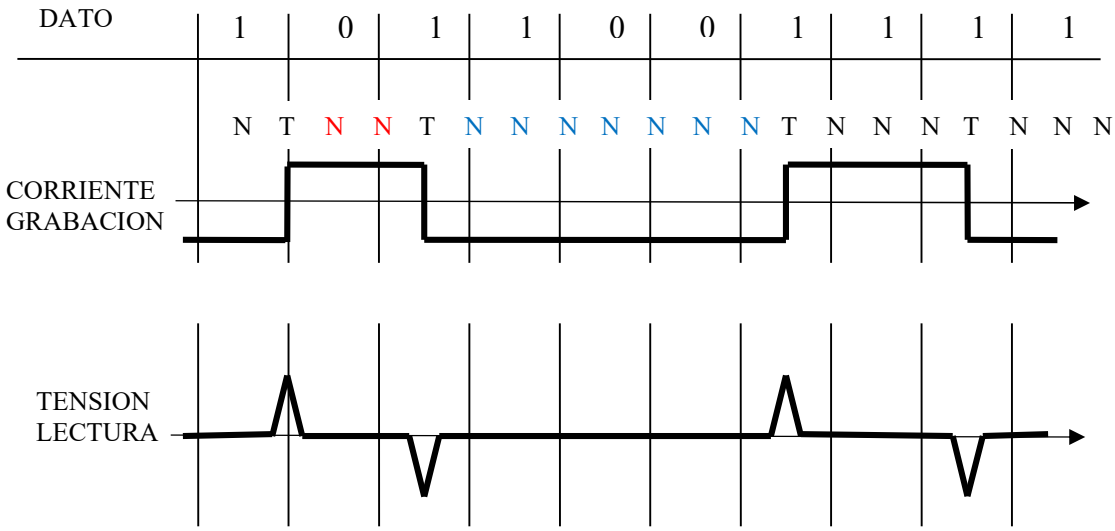


Figura 3.3.11 – Método de grabación RLL 2,7

En la figura se puede ver como se grabaría la misma información de los casos anteriores, con RLL 2,7. Véase que cualquier dato antes o después de esta secuencia respeta la codificación.

ARLL es un nombre que se da al RLL 3,9. Éste tiene una densidad de grabación que duplica al MFM (n=3), sin embargo, por tener menor información de reloj (k=9) tenía problemas de confiabilidad, lo que hizo que fuera abandonado, volviendo al RLL 2,7.

Nota 1: en la actualidad, los discos de alta capacidad codificarían en RLL 1,7, que tiene una menor densidad de grabación que el RLL 2,7 pero mayor confiabilidad. Ahora las altas densidades se logran con la mejora de los materiales y las tecnologías de fabricación.

Nota 2: estas técnicas de codificación de señales se utilizan también en comunicaciones, como emisor de infrarrojos, etc.

Referencias:

Runlength-Limited Sequences. December 1990 - Proceedings of the IEEE 78(11):1745 - 1759
Puede consultar el documento completo en https://www.researchgate.net/publication/2984369_Runlength-Limited_Sequences

3.4 ORGANIZACION Y CARACTERISTICAS

3.4.1 Organización física

Si se considera la recta perpendicular al plato que pasa por las cabezas de lectura/escritura, se observa que la misma genera una superficie cilíndrica al hacerla girar alrededor del eje de rotación. Para cada posición de las cabezas, corresponde un cilindro y la forma de numerarlos es desde el más exterior (cilindro 0) hacia adentro.

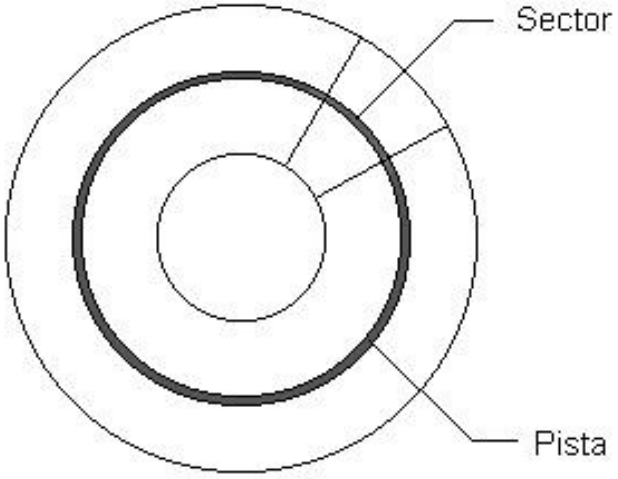


Figura 3.4.1 – Organización física del disco

La intersección de cada cilindro con cada superficie magnética, determina una circunferencia, llamada pista (track). Existen dos pistas por cada placa: una superior y otra inferior.

A su vez, cada pista se divide en arcos determinados por ángulos iguales. Son los sectores, que se numeran desde 1 hasta el último. La mínima unidad direccionable en el disco es el sector, que usualmente contiene 512 bytes de datos.

3.4.2 Formateo

Es el proceso de inicialización del medio de almacenamiento para la unidad de accionamiento. El formateo determina la organización del medio (cantidad de pistas, sectores, bytes por sector) y por lo tanto también su capacidad.

Por ejemplo: un disco rígido de 1985 tenía 4 platos, 850 cilindros, 26 sectores y 512 bytes por sector tendrá una capacidad de:

$$4 \text{ platos} \times 2 \frac{\text{cabezas}}{\text{plato}} \times 850 \frac{\text{pista}}{\text{cabeza}} \times 26 \frac{\text{sectores}}{\text{pista}} \times 512 \frac{\text{bytes}}{\text{sector}} = 90.521.600 \text{ bytes}$$

Para una unidad de cinta con 22 pistas, 8000 sectores y 512 bytes por sector se obtendrá una capacidad de 90.112.000 bytes, es decir aproximadamente 90 Mb.

En cada sector, a su vez, se graba información de control aparte de los datos.

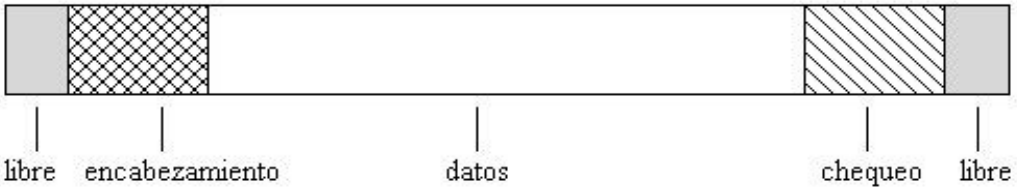


Figura 3.4.2 – Formateo

3.4.2.1 Encabezamiento (header)

El encabezamiento contiene la identificación de la pista la cabeza y el sector. Esto es para que luego del proceso de búsqueda se pueda confirmar el posicionamiento del sistema mecánico. En caso de no coincidir, se informa "error de búsqueda" (seek error).

También posee una secuencia de datos preestablecida para sincronizar la lectura, que consiste en una secuencia de varios ceros y luego un uno, que independientemente del sistema de grabación permite recuperar el sincronismo del reloj, para la posterior lectura de los datos.

El encabezamiento tiene su propio código de verificación.

3.4.2.2 Datos

En la zona indicada como datos es donde residen los datos del usuario (normalmente 512 bytes por sector).

3.4.2.3 Chequeo

La información de chequeo se graba con los datos y luego se verifica al recuperar los datos, para determinar posibles errores de lectura. Algunos algoritmos de chequeo permiten no sólo detectar sino también corregir errores de varios bits.

El método de chequeo usual en medios magnéticos es el CRC (cyclic redundancy check).

El mismo consiste en una función que se aplica a los datos, obteniendo un patrón de bits de longitud definida, por ejemplo 16 bits. La función es una división de polinomios (en módulo 2), donde el dividendo son los datos y el divisor un número fijo, por ejemplo, para el CRC-16 el polinomio, de 17 bits, es $x^{16}+x^{15}+x^{2}+1$. El CRC es el resto de la división.

Para este caso, hay un circuito que calcula el CRC a medida que se graban o leen los datos, lo que lo hace muy práctico para su uso en los medios de almacenamiento (y también muchas otras aplicaciones, como comunicaciones, etc.).

Consiste en un registro de desplazamiento con realimentaciones de compuertas XOR en los FF que corresponden a factores "1" en el polinomio divisor. En este ejemplo, el 74F401 calcula el CRC a medida que la secuencia de datos con su correspondiente reloj ingresa por la línea D (en la parte superior). Cuando el último bit es ingresado, en el registro queda el CRC.

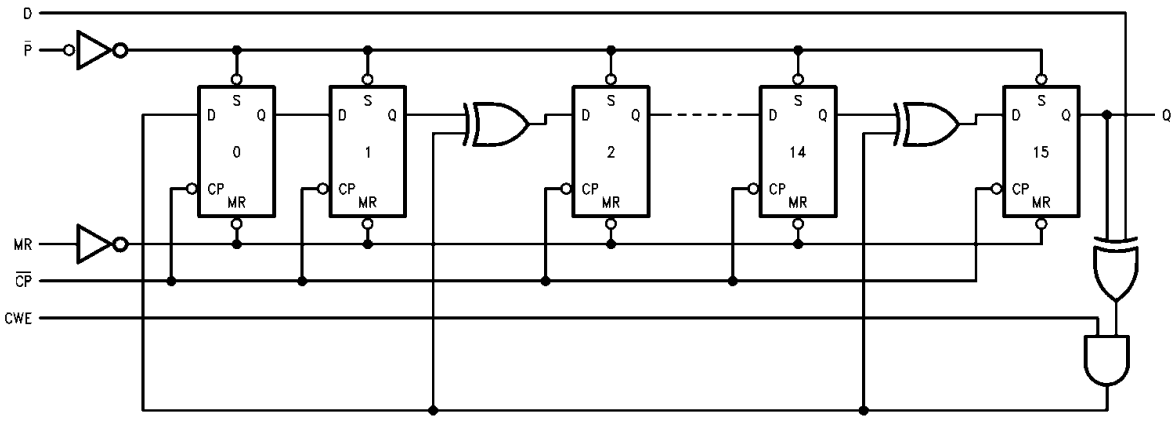


Figura 3.4.3 – Parte del 74F401 – CRC generator/checker

Una propiedad interesante de este circuito es que cuando se realiza la lectura, si el CRC almacenado en el disco se envía a continuación de la secuencia de los datos, el resultado en el registro es "cero" si los datos se corresponden con el CRC. En caso de no ser "cero" se informa "error de CRC".

Es importante hacer notar que el CRC está implementado para detectar los tipos de errores debidos a interferencia electromagnética, rayones, etc, que usualmente afectan muchos bits seguidos y de manera aleatoria.

El CRC-16 se usaba en las unidades de diskettes. En las unidades de discos rígidos se usan CRC de 32 bits, lo que permite, matemáticas mediante, también corregir secuencias erróneas.

Nota:

Una analogía de CRC sería la paridad. Para 16 bits de datos, un bit de paridad permite detectar un error de 1 bit. Con codificación Hamming, para 16 bits de datos, 6 bits de chequeo permiten detectar errores de 2 bits y corregir errores de 1 bit (véase: memorias ECC).

3.4.2.4 Libre (gap)

Los espacios libres (gaps) se dejan para evitar superposición de grabaciones cuando se graba un solo sector sobre una pista previamente utilizada.

Debido a pequeñas diferencias de velocidad de rotación o de la frecuencia de reloj, si no estuvieran los gaps se podría sobrescribir el sector siguiente, inutilizando de esta manera el mismo.

3.4.3 Definiciones

Tiempo de búsqueda

Es el tiempo que tarda el posicionador de la cabeza en ir desde el cilindro origen hasta el cilindro destino.

Latencia

Es el tiempo que tarda la cabeza en encontrar un sector dentro de una pista determinada.

Tiempo de acceso

Es la suma del tiempo de búsqueda y la latencia. Se mide en ms (milisegundos).

Tiempo de acceso promedio

Es la mitad del peor tiempo de acceso. También se puede determinar por métodos estadísticos.

Velocidad de transferencia

Es la velocidad a la que se transfieren los datos entre la unidad de disco y la CPU. Depende de la cantidad de datos que hay en una pista y la velocidad de rotación del plato y se mide en kB/s o MB/s.

3.5 UNIDAD DE DISCOS RIGIDOS

Las placas o platos son de aluminio y están recubiertas con depósitos de metales ferromagnéticos. Las mismas giran juntas, impulsadas por un motor con control de velocidad preciso. En general se trata de un motor sincrónico, con velocidades desde 5400 rpm, para discos de portables o de videovigilancia, de 7200 rpm para aplicaciones generales y excepcionalmente de mayor velocidad para aplicaciones críticas.

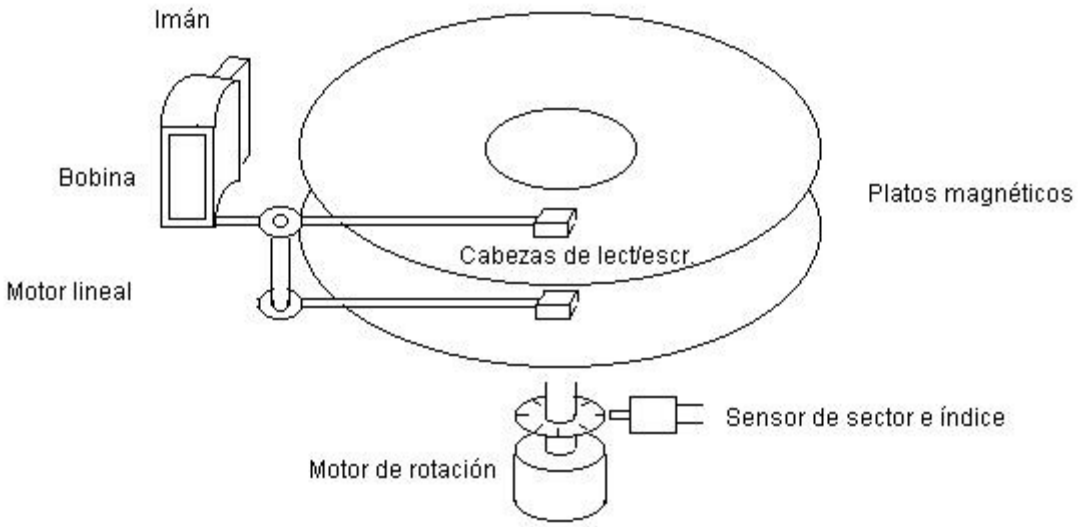


Figura 3.5.1 – Unidad de accionamiento de discos rígidos

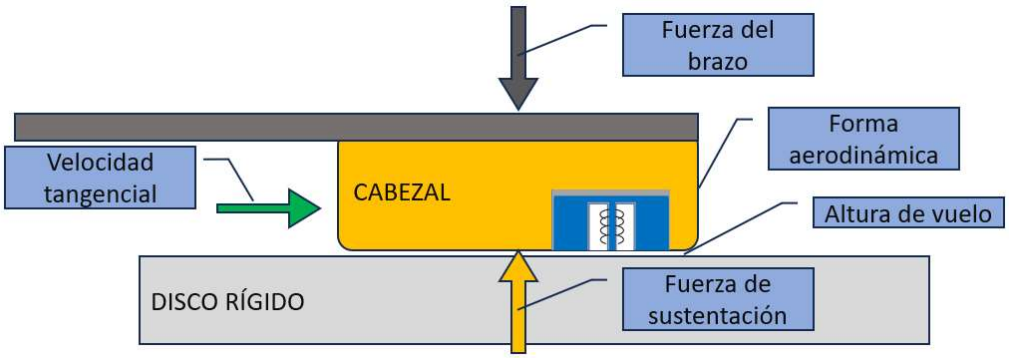


Figura 3.5.2 – Altura de vuelo del cabezal de disco rígido

Cuando el disco está detenido, las cabezas apoyan sobre el plato, usualmente en las pistas interiores (cercanas al eje de rotación).

Cuando el disco gira, las cabezas "vuelan" a pocos micrones de distancia de la superficie. Los factores que afectan a la “altura de vuelo” de la cabeza son: la velocidad angular del disco, la distancia al eje de rotación (componentes de la velocidad tangencial), la forma aerodinámica de las cabezas y la fuerza del elástico que sostiene las cabezas, bastante mayor que el peso de ellas. Debido a estos factores, la cabeza está más lejos del plato en las pistas exteriores que en las interiores.

Figura 3.5.3 – Interior y detalle del motor de accionamiento de las cabezas

El motor que posiciona las cabezas es de tipo lineal (voice coil) para lograr tiempos de búsqueda reducidos. Su principio de funcionamiento es similar al de un parlante de audio, donde una bobina solidaria al soporte de las cabezas está inmersa en un campo magnético creado por un imán permanente fijo al bastidor (chassis) de la unidad. Al circular corriente por la bobina aparece una fuerza perpendicular al plano de la bobina, haciendo que ésta se mueva.

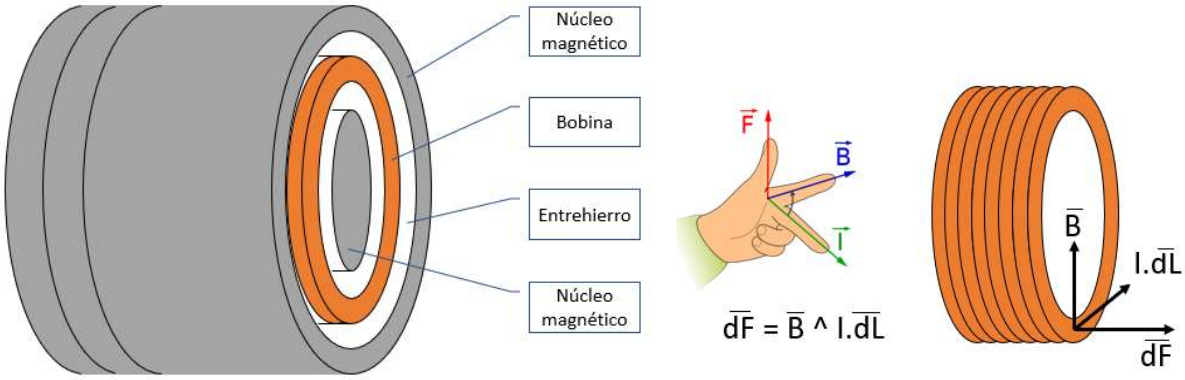


Figura 3.5.4 – Motor lineal

Un sistema realimentado (servosistema) es el encargado de regular la corriente para que la cabeza se dirija al cilindro deseado y una vez allí se mantenga.

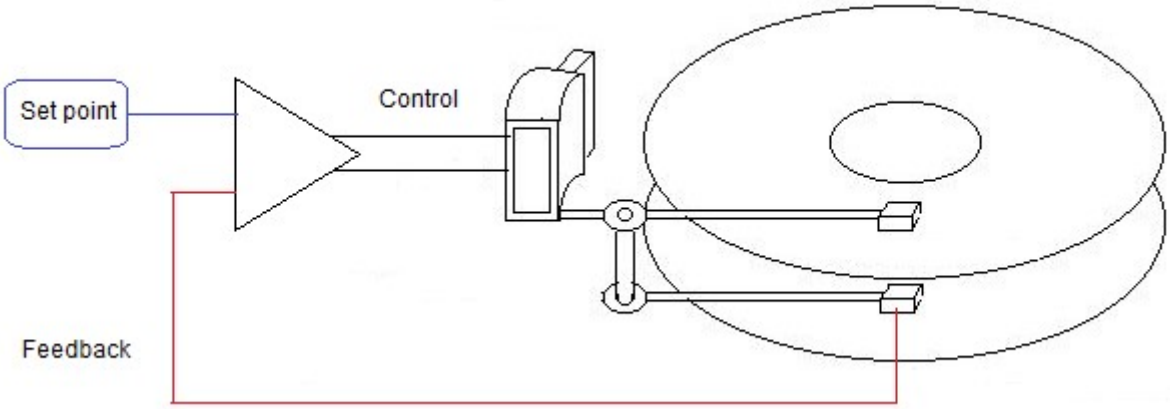


Figura 3.5.5 – Servosistema de posicionamiento

La información que sirve de referencia de posición está grabada en las pistas como se vio en formateo. Existe también otra señal, grabada a ambos lados de cada pista, que sirve para centrar la cabeza en dicha pista.

Figura 3.5.6 – Información de servo

Al comienzo de cada sector, la cabeza se centra tratando de que la señal externa y la interna tengan la misma amplitud. Este sistema tiene la ventaja que, si varían las dimensiones del disco, por ejemplo, por cambios de temperatura, las pistas serán localizadas con total precisión.

Para determinar la posición de los sectores, existe un disco ranurado que, trabajando en conjunto con un sensor inductivo permite la generación de un impulso eléctrico por cada comienzo de sector. Una ranura doble (index), indica el comienzo del primer sector (sector 1). En discos actuales con motor sincrónico se utiliza un sensor Hall para indicar el sector 1 y el resto se hace por temporización precisa controlada por cristal.

Características técnicas de un disco rígido (año 2018)

Barracuda® ST SATA III-6Gb/s 14TB

Specifications	
Model Number	ST14000DM001
Interface	SATA 6Gb/s
Cache	256MB
Capacity	14TB
PHYSICAL	
Height	26.1mm (1.028 in)
Width	101.85mm (4.010 in)
Length	146.99mm (5.787 in)
Weight (typical)	700g (1.543 lb)
PERFORMANCE	
Spin Speed (RPM)	7200 RPM
Sustained data transfer rate	250Mb/s
Average latency	4.16ms
Random read seek time	8.5ms
Random write seek time	9.5ms
I/O data transfer rate	600MB/s
Unrecoverable read errors	1 in 10**15
RELIABILITY	
MTBF	750,000 hours
Annual Failure Rate	0.34%
POWER	
12V start max current	2A
Average wait power	0.8W
Average idle power	5W
Average operating power	6,9W
ENVIRONMENT Ambient Temperature	
Operating	5°–60°C
Nonoperating	-40°–70°C
Maximum operating temperature change	20°C per hour
Maximum nonoperating temperature change	30°C per hour
Shock	
Operating Shock (max)	63 Gs for 2ms
Nonoperating Shock (max)	300 Gs for 2ms
ACOUSTICS	
Acoustics (Idle Volume)	2.8 bels
Acoustics (Seek Volume)	3.2 bels

3.6. UNIDAD DE DISCOS FLEXIBLES

En la figura se muestra un esquema de una unidad de accionamiento de discos flexibles. Las placas magnéticas son de Mylar y están recubiertas en ambos lados con óxidos magnéticos y luego con un material resistente al rozamiento. La placa está contenida en un envase con acceso para las cabezas de lectura/escritura y el aro de arrastre. El tamaño de las placas puede ser de 3,5 y 5,25 pulgadas (actualmente en desuso). Las cabezas rozan sobre la placa cuando se está realizando un acceso. Hay una cabeza de cada lado, que trabaja sobre la respectiva superficie del disco.

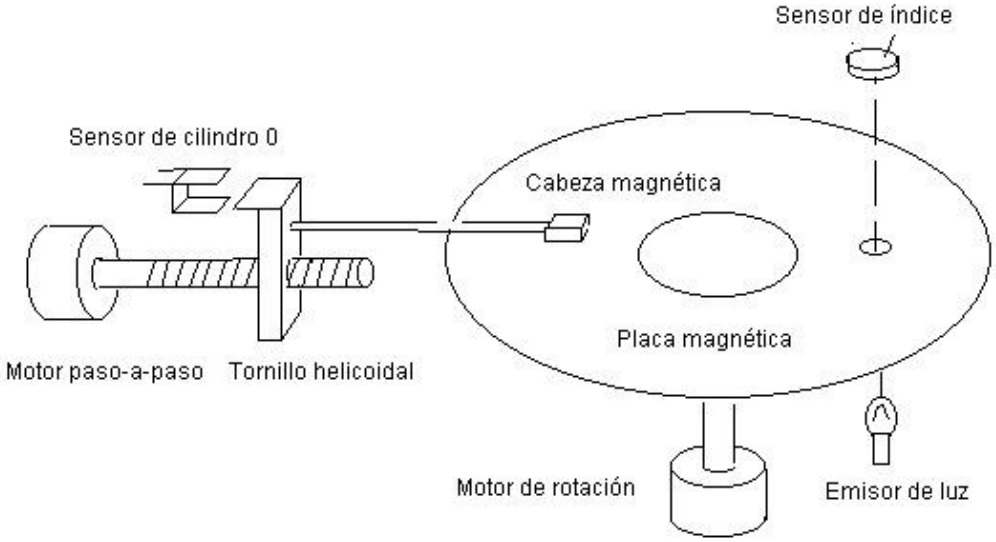


Figura 3.6.1 – Unidad de accionamiento de discos flexibles

El motor que posiciona las cabezas es del tipo paso a paso (stepper motor). La referencia para la posición la da un sensor óptico (sensor de cilindro 0) que es excitado cuando el posicionador de las cabezas retrocede hasta que la lengüeta interrumpe el haz de luz. El cilindro 0 es el más externo, a partir de este, cada paso del motor representa un nuevo cilindro.

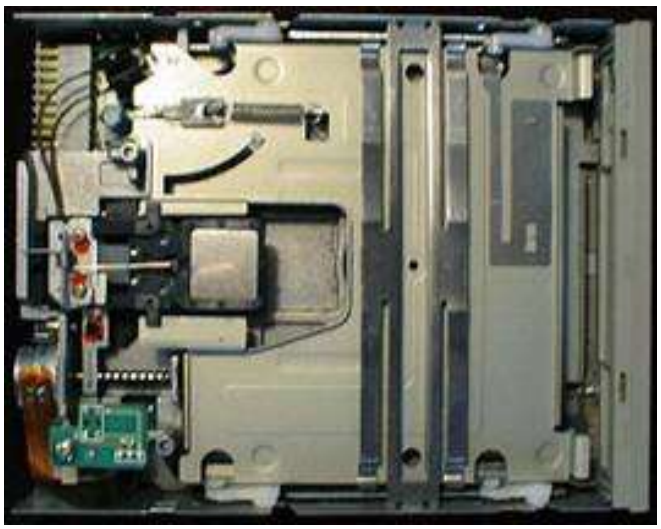


Figura 3.6.2 – Interior de la unidad de accionamiento de discos flexibles

Otro sensor óptico determina la posición del primer sector de un cilindro, gracias a una perforación que posee la placa (en los discos de 5,25 pulgadas). En los discos de 3,5 pulgadas, no es necesario el orificio, ya que el anillo metálico de arrastre posee una muesca que encastra siempre en la misma posición en la unidad de accionamiento.

Los sectores se determinan por software durante el formateo.

3.7. UNIDAD DE CINTA MAGNETICA

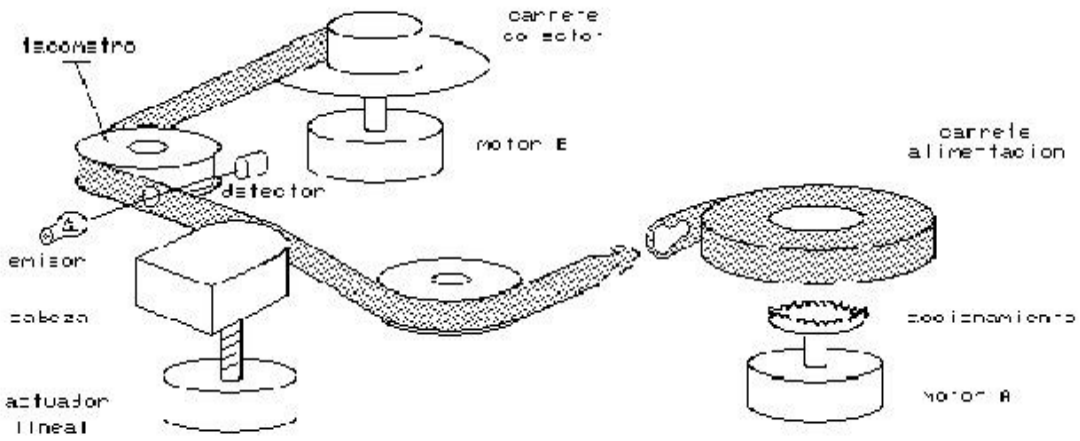


Figura 3.7.1 – unidad de accionamiento de cinta magnética

La unidad de cinta de la figura opera con cassettes de un solo carrete. Las dimensiones de la cinta son ½" (12,7 mm.) de ancho y 600 pies (180 m) de largo y las del cassette 10x10x2 cm. También existen cassettes de dos carretes, como los de audio.

El cabezal posee dos juegos de cabezas: una de grabación y una de lectura para cuando la cinta se mueve hacia adelante y otras dos para el movimiento hacia atrás. La disposición de cabezas de escritura y lectura alineadas y separadas unos 8 mm. permite la operación de verificación leyendo los datos escritos (read-after-write).

El cabezal puede adoptar 11 alturas diferentes gracias a un actuador paso a paso. De esta forma, dado que hay dos juegos de cabezas, se graban 22 pistas.

La combinación de los movimientos de la cinta y de la cabeza determinan pistas en forma de serpentina, que dan origen al nombre genérico de este tipo de unidades (streaming tapes).

Figura 3.7.2 – Disposición de las pistas en una "streaming-tape"

Al introducir el cassette en la unidad, se abre una puerta en la parte posterior de la caja, que permite el acceso al extremo libre de la cinta. Cuando se asegura la traba del cassette también asciende el encastre de accionamiento (motor A).

Un pescante de material plástico adosado al carrete colector, tiene su extremo especialmente diseñado para enganchar el borde libre de la cinta y tirar (mediante el motor B)

hasta que la misma quede enhebrada en ambos carretes. En este momento el cassette no puede ser retirado porque se encuentra accionada la traba de seguridad, manejada por un electroimán.

La cinta posee un orificio que activa el detector (ver figura) y le indica a la unidad que se alcanzó el comienzo de cinta. La velocidad de la cinta se controla regulando las velocidades de los motores A y B (de corriente continua) y teniendo en cuenta la información que suministra un generador de pulsos (tacómetro) solidario a una de las poleas. La tensión de la cinta se ajusta en función del consumo de los motores.

Este tipo de unidades, usualmente está controlada por microprocesador, por lo tanto posee diagnósticos internos, tanto de la parte electrónica como de la mecánica y brinda al controlador información completa de sus condiciones operativas.

La descripción precedente corresponde a una unidad de cinta que se usaba con las mini-computadoras DEC PDP-11, para realizar backup de los discos rígidos (1985).

En la figura siguiente se puede apreciar una unidad actual y su correspondiente medio. Estos medios pueden ser de lectura escritura o WORM (Write Only Read Many). Esta última se utiliza con fines de almacenamiento en aplicaciones donde los datos no puedan ser alterados una vez grabados, por ejemplo por temas legales o de seguridad informática.

Figura 3.7.3 – Cinta magnética y unidad de accionamiento de cinta

Características técnicas de una unidad de accionamiento de cinta (2011)

TS1140 tape drive

Characteristics	
Recording technique	Linear Serpentine
Number of tracks	2560
Native capacity (uncompressed)	4 TB (using JC/JY media), 1.6 TB (using JB/JX media), or 500 GB (using JK media)
Native sustained data rate (uncompressed)	250 MBps
Burst data rate	800 MBps
High-speed search (max)	12.4 mps
Warranty	One year
Physical characteristics	
Dimensions	95 mm H x 198 mm W x 467 mm D (3.8 in x 7.8 in x 18.4 in)
Weight	5.7 kg (12 lbs 7 oz)

3.8. UNIDAD DE DISCOS ÓPTICOS

3.8.1 Unidad de CD ROM, DVD ROM y Blu-ray

El medio, removible y de bajo costo, es un disco de policarbonato de 12 cm de diámetro en el que se graba una sola pista en forma de espiral, comenzando en el interior del mismo. La grabación se realiza colando el policarbonato en un molde o matriz que tiene la forma complementaria a la que se desea obtener.

Figura 3.8.1 – Pista en espiral

La unidad de discos ópticos utiliza un láser para la lectura de la información. Un sistema óptico dirige la luz del emisor hasta la superficie del disco y simultáneamente, a través de prismas, al detector. La luz reflejada en el disco también es dirigida al receptor, como se ve en la figura:

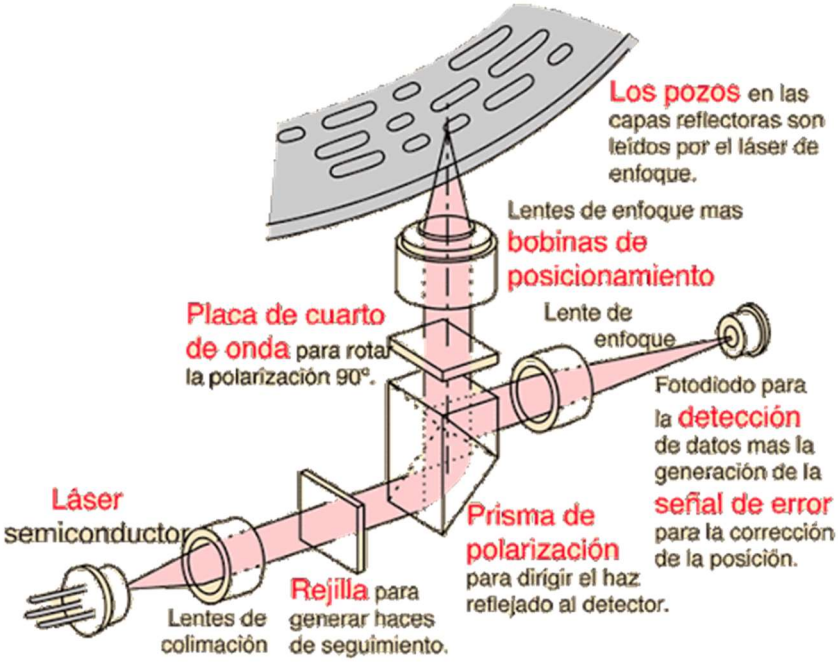


Figura 3.8.2 – Unidad de CD

Para que el sistema de lectura pueda recorrer toda la pista, se cuenta con un motor que mueve el cabezal hacia adentro o hacia afuera del eje de rotación.

El movimiento del cabezal es acompañado por unas bobinas que mueven la lente para "saltar" de pista, manteniéndose siempre enfocado a donde debe leer. La velocidad de rotación

del eje es variable, de manera de mantener una velocidad lineal constante del cabezal respecto de la pista espiral. Esto se debe a que por diseño se pretende una tasa de bits constante en toda la superficie del disco.

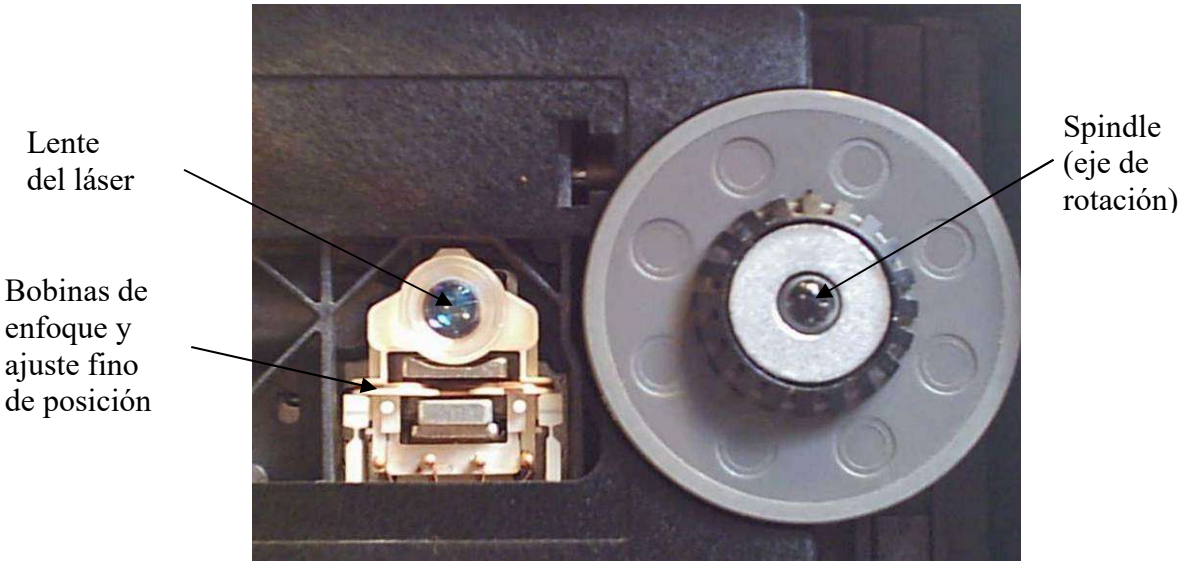


Figura 3.8.3 – Detalles de una unidad de accionamiento

Los datos se codifican mediante pozos (pits) y planos (lands) en la superficie del disco. La profundidad de un pozo (respecto del plano) corresponde a ¼ de la longitud de onda de la luz utilizada. De esta forma, como se mencionó anteriormente que tanto la luz emitida como la reflejada llegan al detector, ambas ondas se suman o restan dependiendo de si han incidido en un pozo o un plano, como muestra la figura siguiente:

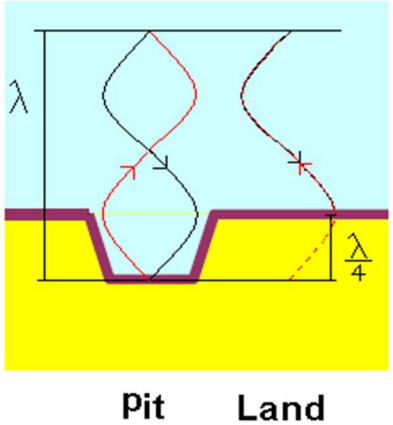


Figura 3.8.4 – lectura de un CD ROM

Así, los pozos se "ven" como "oscuros" (las ondas se restan) y los lands como "claros" (las ondas se suman). De esta manera al leerse la pista se pueden detectar las transiciones entre pozos y planos lo que asociado a la temporización permite recuperar la información digital (como hacíamos en los medios magnéticos).

En el caso de los CD el método de codificación utilizado es EFM (Eight to Fourteen Modulation), una derivación de RLL que codifica 8 bits en 14 combinaciones de T (transición) y N (no transición).

Es de destacar que en ningún caso hay pozos o planos de menos de 3 ciclos de reloj, lo que facilita la lectura y tampoco de más de 10 ciclos de reloj lo que facilita la recuperación del reloj.

Los DVD utilizan el método EFMPlus, derivado del anterior y con mayor densidad de información. Ambos fueron patentados por Philips.

El formateo de los sectores con códigos de verificación permite la corrección de errores en ciertos casos, como rayones o huellas en la superficie del disco.

Referencias:

Runlength-Limited Sequences . December 1990 - Proceedings of the IEEE 78(11):1745 - 1759
Puede consultar el documento completo en https://www.researchgate.net/publication/2984369_Runlength-Limited_Sequences

Características de los medios

En las siguientes figuras se aprecian las características comparativas entre un CD, un DVD y un Blu-ray. Nota: NA (apertura numérica de la lente).

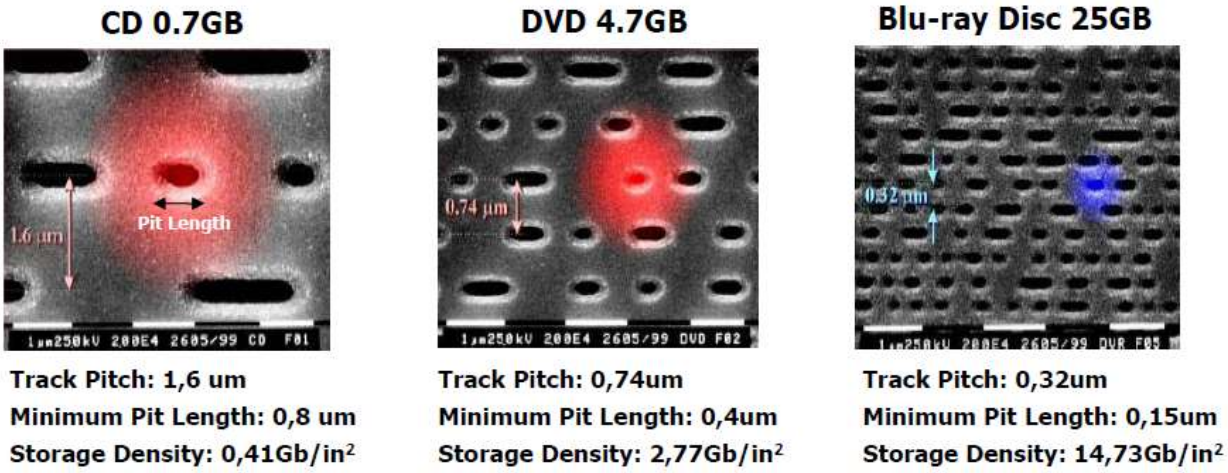


Figura 3.8.5 – Densidad de almacenamiento. Comparativa entre CD, DVD y Blu-ray

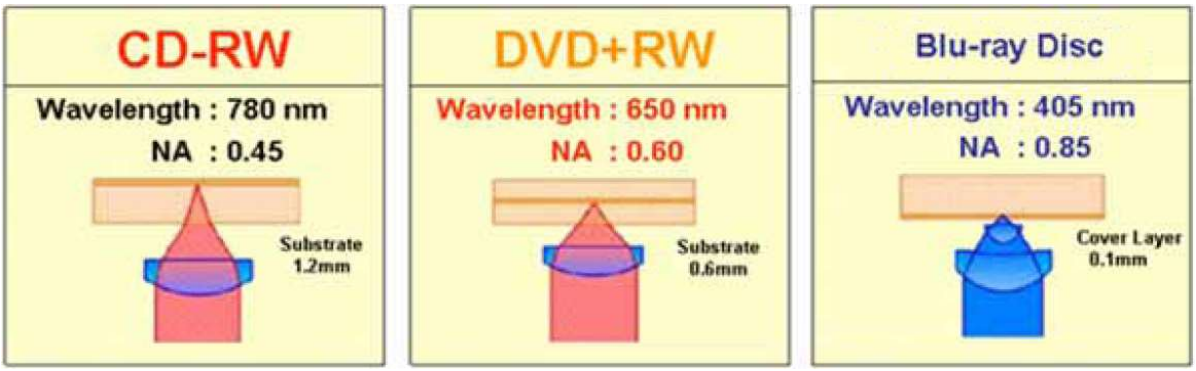


Figura 3.8.6 – Longitud de onda. Comparativa entre CD, DVD y Blu-ray

3.8.2 Unidad RW (Read/Write)

Cuando los discos RW se fabrican, el área reescribible es tal que hay surcos a lo largo de una pista espiral continua. La superficie de estos surcos es plana, con una capa reflectiva (es decir, la superficie de grabación está en ella es estado cristalino, significando que no hay datos).

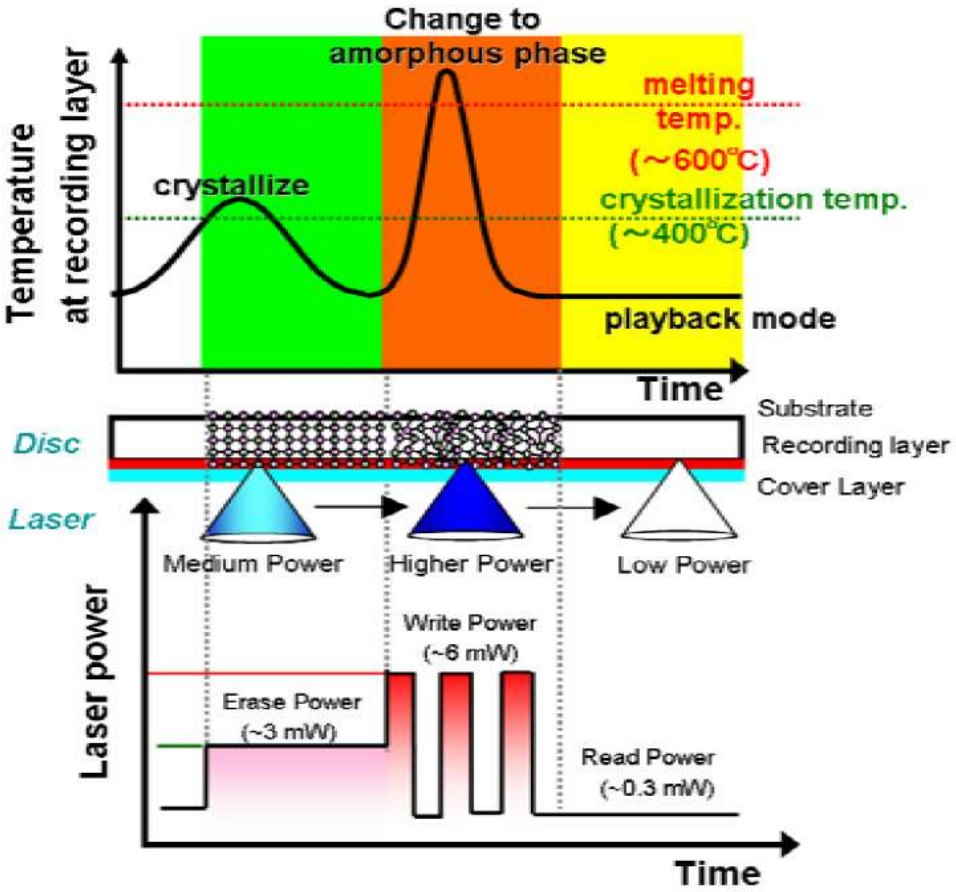


Figura 3.8.7 - Proceso de grabación de cambio de fase

La capa de grabación del disco RW se hace de un material que pueda cambiar su estado (o fase) a partir de una fase cristalina (land) a una fase amorfa (pit). La combinación de segmentos cristalinos y amorfos constituye una sección grabada.

Cuando se graba en el disco, la potencia del láser de la unidad de grabación es controlada de manera que varíe entre el punto de la cristalización, el punto de fusión, y un punto de enfriamiento, que es cercano a la temperatura ambiente. Controlando la potencia del láser de esa forma mientras el disco gira, se crean las secciones grabadas.

Referencias:

Blu-ray Disc Asocciation - 1.C Physical Format Specifications for BD-ROM 5th Edition - March, 2007. Puede consultar el documento completo en http://www.blu-raydisc.com/Assets/Downloadablefile/BD-ROMwhitepaper20070308-15270.pdf.

3.9. ARREGLOS DE DISCOS (RAID)

El acrónimo RAID (Redundant Array of Inexpensive Disks) aplica en forma genérica a una combinación de dos o más discos rígidos para mejorar las prestaciones de velocidad, confiabilidad o ambas. La idea original (y de allí el nombre) fue lograr la confiabilidad de un disco costoso con dos discos económicos, grabando los mismos datos en ambos. Luego aparecieron más combinaciones y se generó la clasificación RAID de un conjunto de discos.

Referencias: http://es.wikipedia.org/wiki/RAID.

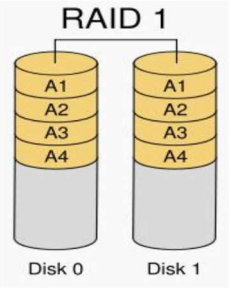


Figura 3.9.1 – RAID 1

El RAID 1 o espejado de discos aumenta la confiabilidad replicando los mismos datos en dos discos idénticos. La capacidad y la velocidad del conjunto es la misma que la de cada componente por separado.

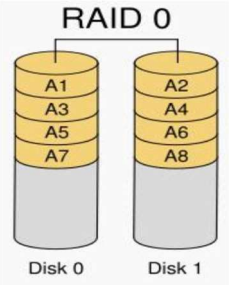


Figura 3.9.2 – RAID 0

El RAID 0, agrupa dos discos como si fuese uno solo de mayor capacidad. Esta configuración en teoría aumenta la velocidad y evidentemente reduce la confiabilidad, ya que la falla de cualquiera de ellos perjudicará al conjunto.

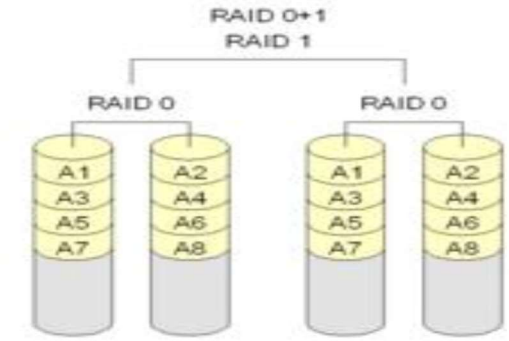


Figura 3.9.3 – RAID 0+1

RAID 0+1 es una combinación de los dos anteriores. Se trata de agrupar dos juegos RAID 0 en una configuración RAID 1. La confiabilidad del conjunto es igual a la de cada componente, se logra un incremento de velocidad y la capacidad es igual a la de cada conjunto RAID 0. Ejemplo: si se usan cuatro discos de 160 GB, se obtendrá un conjunto de 320 GB que la computadora ve como un disco único.

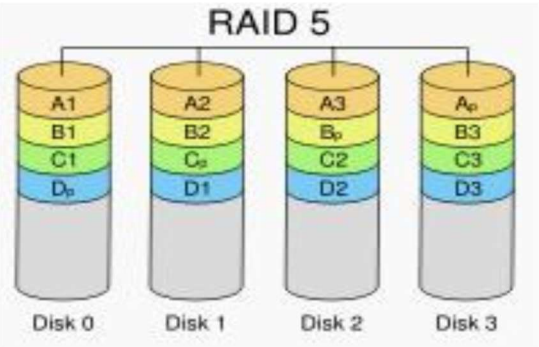


Figura 3.9.4 – RAID 5

RAID 5 requiere al menos tres discos. En el ejemplo de la figura se usan cuatro. Los archivos se dividen en “stripes” (A1, A2, A3) que se graban en discos diferentes y también se calcula y graba en otro disco la paridad (Ap). Si uno de los discos falla, los datos que estaban en él se pueden reconstruir a partir de la información de paridad guardada en los demás. El disco averiado se puede así reemplazar, logrando un aumento de confiabilidad con una reducida pérdida de capacidad de almacenamiento del conjunto.

Figura 3.9.5 – Unidad de almacenamiento con tecnología RAID

3.10. CONFIABILIDAD

En el apartado anterior se vio que, según la configuración utilizada, con dos discos se podía aumentar o disminuir la confiabilidad.

Definimos la confiabilidad como la inversa de la tasa de fallas.

Supongamos para cada disco rígido una tasa de fallas (MTBF) de 1 cada 3 años, lo que equivale aproximadamente a 1 cada 1000 días. La probabilidad de falla es entonces 0,001.

Si los discos están configurados como RAID 1, es decir en espejo, los mismos datos están en ambos discos y la probabilidad de perder los datos a causa de una falla interna del disco estará dada por el producto de las probabilidades de falla de cada uno, dado que son sucesos independientes, entonces:

P(RAID 1) = P(DISCO 1) * P(DISCO 2) = 0,001 * 0,001 = 0,000001

Esta fórmula es la misma que se usa para determinar la probabilidad de que salga 2 como suma de dos dados. La probabilidad de obtener 1 en cada dado es 1/6 y en los dos dados 1/36.

El valor 0,000001 es muy pequeño, tanto que se corresponde con 1 en 2700 años.

Si, por el contrario utilizamos los discos en RAID 0, entonces cualquiera de ellos que falle hará fallar al conjunto, entonces:

P(RAID 0) = P(DISCO 1) + P(DISCO 2) = 0,001 + 0,001 = 0,002

Dicho de otro modo, en el mismo período tendremos el doble de fallas, dado que tenemos el doble de discos en uso. El valor 0,002 se corresponde con 1,5 años aproximadamente.

Estos cálculos están muy simplificados y merecen algunas consideraciones:

- El tiempo de reparación (MTTR) debe ser muy bajo en relación al MTBF, por ello los sistemas RAID poseen discos que se pueden reemplazar sin apagar el sistema.
- El aviso de que un disco está en falla debe ser inmediato, ya que de no hacerlo se estará incrementando el MTTR. Si no se cuenta con este aviso, el sistema podría estar trabajando con un disco solo y por ende su confiabilidad bajaría drásticamente, al valor del disco simple.

En las configuraciones mas complejas, se aplican las mismas reglas de cálculo de probabilidades, por ejemplo en el RAID 0+1 tenemos:

P(RAID 0+1) = P(RAID 0) * P(RAID 0) = (P(DISCO 1) + P(DISCO 2)) * (P(DISCO 3) + P(DISCO 4)) = (0,001 + 0,001) * (0,001 + 0,001) = 0,002 * 0,002 = 0,000004

Este conjunto tiene una confiabilidad excelente, sumado a un aumento de la velocidad de transferencia al doble y una capacidad total igual a la mitad de la suma de todas las capacidades.

Si los discos son de 1 TB de capacidad y 100 MB/s de velocidad de transferencia sostenida, el conjunto RAID 0+1 tendrá una capacidad de 2 TB y una velocidad de 200 MB/s.

La tasa de fallas de 0,000004 se corresponde con 1 en 685 años.

Estas técnicas de redundancia para aumentar la confiabilidad se usan profusamente en aviación, centrales nucleares y procesos industriales críticos. En lo relacionado con procesamiento de datos, se usa esta técnica en los servidores de datacenters, etc.

En los equipos de medición y control se aplican los discos “espejados” o RAID 1 dado que en general son las únicas partes móviles del sistema y por ende las más propensas a fallas.

CAPÍTULO 4

ARQUITECTURA DE COMPUTADORAS PERSONALES

4.1 INTRODUCCION

La primera PC desarrollada en torno a los microprocesadores de Intel 80X86 fue la IBM PC. Nacida en 1981, IBM adoptó para la PC el microprocesador 8088 como unidad central de proceso. La característica más destacable de estos equipos fue su concepción "abierta" que permitía ampliar o agregar nuevas prestaciones con gran facilidad. Esta característica se conserva aún hoy en los nuevos modelos, generándose una norma no escrita, conocida como "compatibilidad IBM".

A la IBM PC sucedió en 1983 la XT, que conserva el microprocesador 8088, pero con el agregado de algunas mejoras, como la aparición del disco rígido.

El fenómeno de la "compatibilidad" no sólo se hizo visible en la aparición de una gran cantidad de adaptadores y una variedad de software de aplicación, sino también en la aparición de réplicas semejantes a los modelos de IBM denominadas "compatibles", fabricadas por otras empresas como NEC, Compaq, Dell, Samsung, etc. y las PC sin marca o "clones".

4.2 COMPONENTES DEL SISTEMA

Si observamos el diagrama en bloque de la PC XT, identificaremos varios de los componentes vistos a lo largo del curso. Podemos ver el microprocesador 8088, el coprocesador matemático 8087, el controlador de interrupciones 8259, el de DMA 8237, etc.

También se pueden observar los conectores de expansión del BUS, que le dan a la PC la característica de arquitectura "abierta", ya que permite el diseño de cualquier tipo de módulo de entrada salida, conectable al mencionado BUS.

A continuación, describiremos cómo se interconectan los mencionados componentes y las funciones principales de cada uno de ellos.

4.2.1 Microprocesador 8088

Este uP contiene el mismo set de instrucciones que el 8086, sólo que posee un BUS de datos de 8 bits que lo hacía compatible con los dispositivos diseñados para el anterior microprocesador de Intel 8085. Por este motivo el Bus de datos de la PC XT es de solo 8 bits, aunque el funcionamiento interno del microprocesador es de 16 bits.

En la primera versión de la PC XT, el microprocesador funcionaba a una frecuencia de reloj de 4,77 MHz, motivo por el cual se utilizaba un oscilador de frecuencia triple de 14,318 MHz. Versiones posteriores de PC incorporaron chips de mayor velocidad, como el caso del NEC-V20 que posee las mismas características que el 8088 pero funcionaba a 8 MHz.

4.2.2 Coprocesador matemático 8087

Junto al microprocesador es común encontrar un zócalo de 40 pines para la instalación del circuito integrado 8087, que es la unidad de punto flotante (FPU) diseñada para el 8086/8088 (también llamada unidad de procesamiento numérico NPU). Este coprocesador agrega instrucciones numéricas y registros de punto flotante, lo que incrementa notablemente la velocidad de procesamiento, en especial cuando los programas requieren operaciones numéricas de alta precisión.

El agregado de esta opción sólo requiere la instalación del coprocesador en el zócalo y la habilitación de un switch en la placa madre, que indica la existencia del coprocesador. Este switch permite que el microprocesador reciba la interrupción de excepción del coprocesador.

4.2.3 Controlador de interrupciones 8259

Otro de los chips de la familia Intel que integran la PC es el controlador programable de interrupciones (PIC) 8259. Este chip permite ampliar a ocho las entradas de interrupción, generándose las señales llamadas IRQ0 a IRQ7. El diseño de la placa madre de la PC prevé la asignación de algunas de éstas para funciones específicas, tal es el caso de las IRQ0 e IRQ1. La primera de ellas se conecta a la salida del contador del timer y sirve para proveer una base de tiempo confiable. En la programación inicial del 8259 se le asigna a esta entrada el vector de interrupción 8, por los que las restantes poseen los vectores siguientes (9 a 15). La IRQ1 (vector 9) se emplea para la lectura del teclado.

Las restantes interrupciones se encuentran en el slot de expansión del BUS, lo que permite que sean usadas por cualquier dispositivo conectado en dicho lugar. Sin embargo, la estandarización del hardware y software mencionada con anterioridad hizo que los controladores más comunes conserven una interrupción como predeterminada. Esto implica que las mismas no deben ser utilizadas por otros dispositivos cuando se encuentre instalado alguno de esos controladores. A continuación, se proporciona una lista de las interrupciones y se mencionan los controladores más comunes que hacen uso de ellas.

Interrupción	Aplicación
IRQ2	RESERVADA
IRQ3	COM/SDLC
IRQ4	COM/SDLC
IRQ5	DISCO DURO
IRQ6	DISCO FLEXIBLE
IRQ7	LPT

Tabla 4.2.1 – Interrupciones de la PC

La IRQ2 que figura como reservada permite que se la utilice para agregar un segundo chip 8259 en cascada, lo que implica un aumento de 7 interrupciones. Esto se incorporó a la placa madre de la PC-AT que sucedió a la PC, e incorporaba el microprocesador 80286 como unidad central de proceso.

4.2.4 Controlador de DMA 8237

También podemos observar el controlador 8237, que proporciona cuatro canales de DMA para dispositivos de entrada salida que requieran de este servicio. Estos canales de denominan con las siglas DMA0 a DMA3, y al igual que las interrupciones se encuentran preasignados para cumplir con las funciones básicas. El canal 0 se emplea para realizar el refresco de la memoria RAM dinámica, por lo que se conecta directamente al timer del sistema. Esto permite que se realice periódicamente cada intervalo de tiempo definido para evitar que el contenido de la memoria se desvanezca por completo.

Los restantes canales se encuentran disponibles en el conector de expansión con sus señales DRQ y DACK para que puedan ser usados por los dispositivos de entrada salida que lo requieran. A continuación, se proporciona una lista de la asignación de los canales en la PC.

Canal de DMA	Aplicación
CANAL 0	REFRESCO DE MEMORIA
CANAL 1	SDLC
CANAL 2	DISCO FLEXIBLE
CANAL 3	NO ASIGNADO

Tabla 4.2.2 – Canales de DMA de la PC

4.2.5 Timer 8253

Este chip es un contador/timer que provee a la PC de 3 timers programables utilizados para todas las mediciones de tiempo. Algunas de las aplicaciones ya fueron mencionadas cuando se describieron los controladores 8259 y 8237, aunque no fueron completamente analizadas.

El 8253 provee tres canales independientes, cada uno de ellos programable de 6 modos diferentes. Las entradas de reloj pueden ser distintas para cada canal, aunque en la PC es única y de 1,19 MHz.

4.2.6 Rom BIOS

Se mencionó que existe un hardware estándar que compone la base de la PC. Asimismo, con la primera PC, IBM proveyó también un programa estándar de control para muchos de los componentes de este hardware. Este programa reside en la ROM del sistema y se lo llama comúnmente ROM BIOS (Basic Input/Output System).

Esta interfaz permite que los diseñadores de software de aplicación puedan realizar programas que funcionen en sistemas de configuración variada, sin tener que realizar modificaciones para cada caso. De esta forma también los fabricantes pueden realizar modificaciones en el hardware, pero manteniendo esta interfaz tal cual es, no se necesitan modificar los programas. Igualmente, versiones posteriores de BIOS incorporan nuevas funciones, sin perjuicio de las anteriores. Esto posibilita que los programas viejos puedan ejecutarse correctamente, aunque por supuesto no harán uso de estas nuevas funciones.

La ROM BIOS posee tres funciones específicas:

La primera de ellas se denomina POST (Power On Self Test). El POST consiste en una rutina que se ejecuta cuando se enciende la computadora o cada vez que se realice un Reset. La función principal de esta rutina es testear el hardware e inicializar los dispositivos programables.

Otra de las funciones es contener los drivers de los dispositivos de I/O. Tanto la versión de IBM como las posteriores proveen una gran variedad de rutinas para el hardware usado más comúnmente.

La tercera función del BIOS es contener un set de servicios del sistema, que si bien no controla ningún dispositivo, proveen datos valiosos acerca de la operación del sistema.

4.2.7 BIOS Setup

Normalmente se accede al menú de Setup oprimiendo "supr" o "esc" o "F10" según la PC de que se trate y aparece la pantalla en modo texto en los BIOS antiguos o en modo gráfico en las más nuevas.

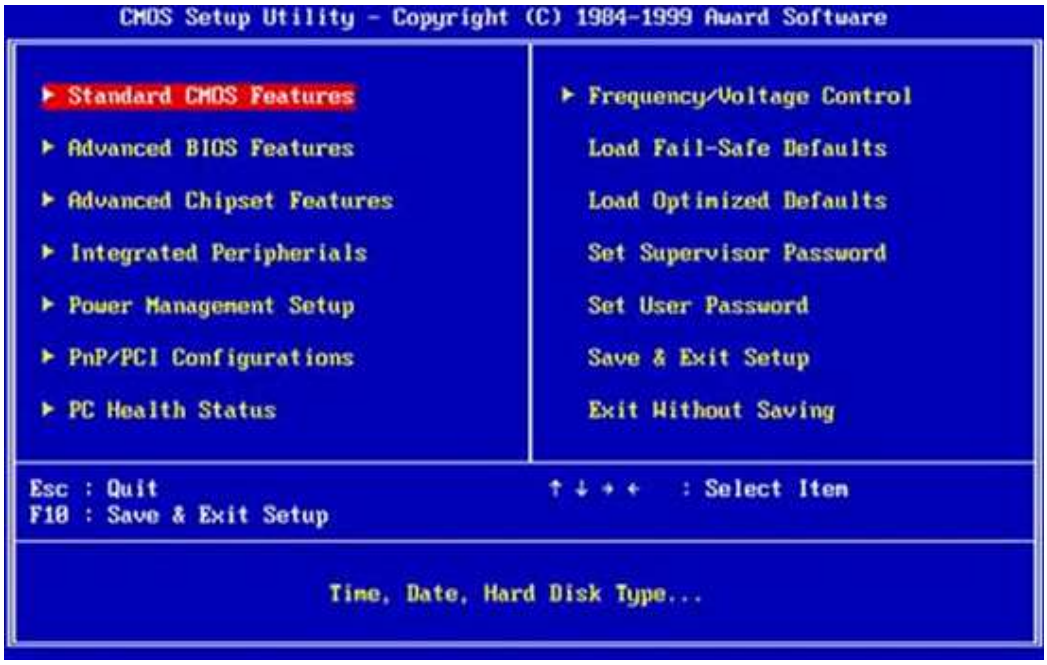


Figura 4.2.1 – Interfaz BIOS (Basic Input Output System)

4.2.8 Diagrama en bloques

En la PC los canales se encuentran asignados como se muestra en la figura.

Como se puede ver, el canal 0 se encuentra siempre habilitado y se utiliza como contador del sistema para proveer una base de tiempo confiable. La salida del timer proporciona una interrupción, comúnmente llamada timer tic, que se conecta a la IRQ0. La misma es utilizada por el BIOS para incrementar un contador de 4 bytes en 0040:006C que se utiliza para el cómputo de la fecha y la hora. El canal 1 está también siempre habilitado y programado para una cuenta de 15uS, pero su salida se conecta al canal 0 de DMA. El final de cuenta de este timer dispara un ciclo de lectura de memoria ("dummy") que permite que no transcurra demasiado tiempo sin que ésta sea leída. Este ciclo conocido como refresco de memoria es imprescindible para que el contenido no se desvanezca. Por último el canal 2 se encuentra disponible, aunque como se pueda ver se encuentra conectado al altavoz. Si se quisiera usar con otros fines, se debe

deshabilitar el parlante a través de la salida B1 del 8255, y se podrá habilitar el timer a través de B0. El final de cuenta puede detectarse por la misma 8255, a través de la entrada C5.

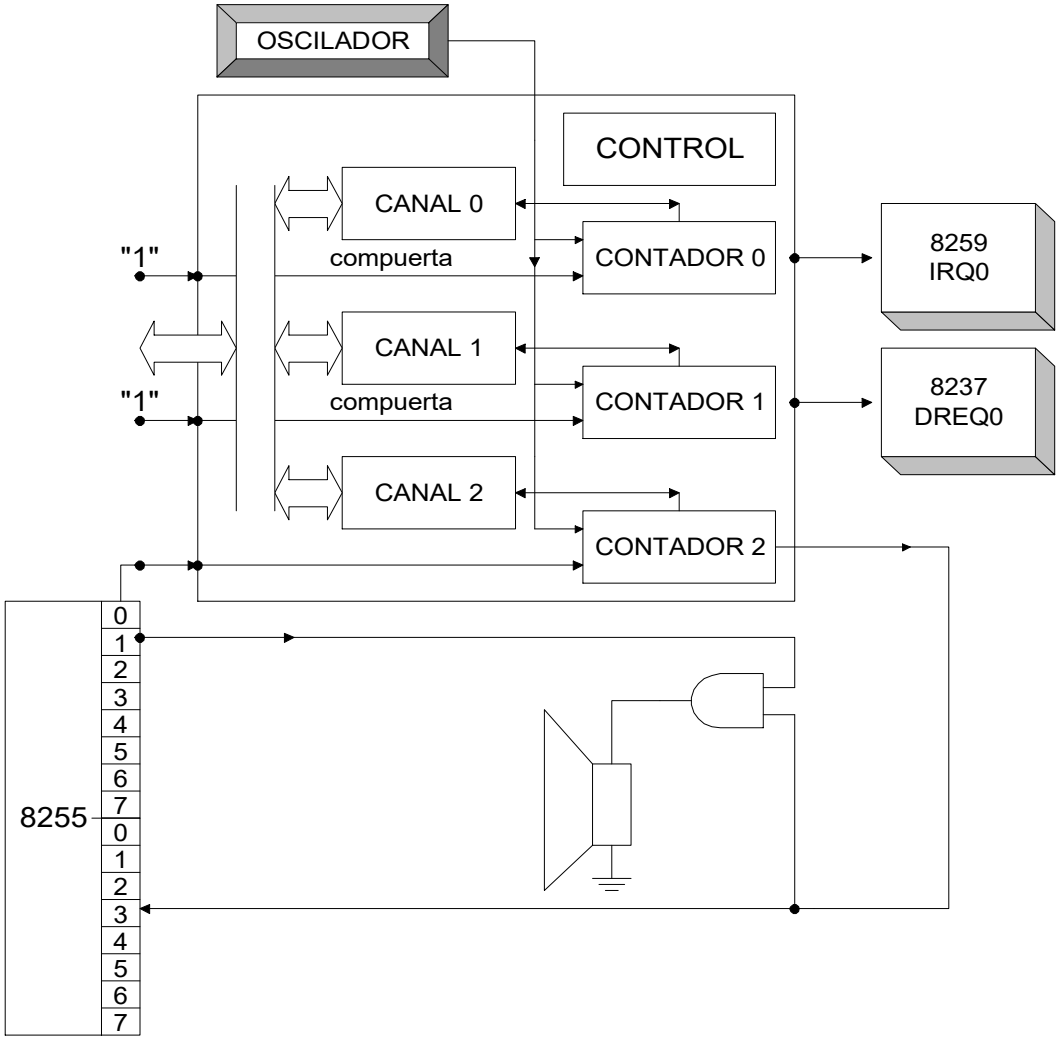


Figura 4.2.2 – Diagrama en bloques de la PC original

4.3. BUSES EN LA ARQUITECTURA PC-COMPATIBLE

4.3.1 El bus ISA

El bus ISA *(Industry Standard Architecture),* fue el primer bus utilizado en los equipos PC y XT de IBM como sistema de arquitectura abierta. Esto permitió en gran parte la proliferación de clones y la gran variedad de tarjetas de expansión e interface de bajo costo que aún se encuentran en el mercado. De los buses ISA hay dos versiones: el de *8 bits* y el de 16 *bits.*

El bus ISA de 8 bits

El bus ISA de *8 bits* salió al mercado en el año 1982 con el PC y luego se conservó en los XT's. Este bus se encuentra en forma de conector en línea de doble lado y tiene 62 contactos, 31 por cada lado. Sus pines o contactos están numerados por un lado como A1 hasta A31 y por el otro lado como B1 hasta B31. En la figura 4.3.1 se muestra la estructura física, la numeración y el nombre de cada una de las señales.

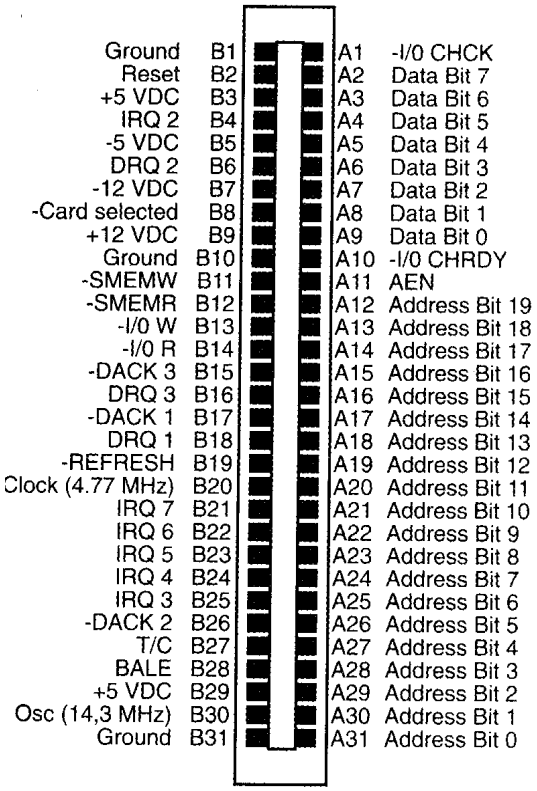


Figura 4.3.1 – Bus ISA de 8 bits

Como se puede observar en la figura, en una ranura o *slot* se encuentran el bus de datos, el bus de direcciones, las señales de control y de reloj y los voltajes de alimentación. Casi todas estas señales van conectadas al microprocesador a través de circuitos acopladores *(drivers o buffers)* con el fin de protegerlo de posibles cortocircuitos o conexiones equivocadas. Como podemos ver, en el bus ISA de *8 bits* hay un bus de datos de *8 bits* (de ahí su nombre) y un bus de direcciones de 20 líneas (lo que permite un direccionamiento hasta de *1 MByte.*

El bus contiene seis señales de interrupción (IRQ2 a IRQ7), tres canales de DMA y una señal de reloj de 4.77 MHz. Aunque cada conector en el bus se supone que trabaja de la misma forma, los primeros PC's fabricados con ocho ranuras requerían tener por lo menos una tarjeta conectada en el *slot 8* (el más cercano a la fuente de poder), con el fin de obtener una señal llamada *"card selected (tarjeta seleccionada)"* en el pin B8. Esta señal era una línea diseñada para recibir un adaptador especial de IBM llamado el 3270PC, pero la mayoría de los fabricantes de clones y compatibles no siguió esta configuración.

El pin del oscilador (B30), entrega una señal de 14.3 MHz. Cuando se debe hacer reset en el PC, el pin RESET DRV (B2) reinicia todo el sistema. Cuando se recibe una dirección válida, el pin AEN (A11) le indica al sistema que se puede decodificar esta dirección. El pin -I/O CHCK o I/O Channel Check (A1), avisa a los circuitos de la tarjeta principal (motherboard) que ha ocurrido un error en la tarjeta de expansión; el signo negativo indica que es una señal activa baja. El pin -I/O CHRDY o I/O Channel Ready (A10) se activa cuando una tarjeta de expansión está lista. Si este pin está en nivel bajo (0), el microprocesador extiende el ciclo del bus generando estados de espera (wait states).

Las seis señales de interrupción por hardware (IRQ2 a IRQ7), se utilizan por las tarjetas de expansión para demandar atención por parte del microprocesador. Las interrupciones 0 y 1 no están disponibles en el bus ya que ellas tienen las prioridades más altas del temporizador principal y el teclado. Las señales I/O Read y I/O Write indican que el microprocesador o el controlador de DMA quieren transferir datos hacia o desde el bus de datos. Las señales de lectura y escritura de memoria (-MEMR y -MEMW) le indican a la tarjeta de expansión que la CPU o el DMA van a leer o escribir datos a la memoria principal. El bus XT tiene tres señales de requisición de DMA (DRQ1 a DRQ3), que le permiten a la tarjeta de expansión transferir datos hacia o desde la memoria.

El bus ISA de 16 bits

Con el avance de la tecnología, las limitaciones del bus ISA de *8 bits* se evidenciaron rápidamente. Las seis interrupciones disponibles se coparon con la unidad de disco flexible, el disco duro, los puertos seriales y el puerto paralelo, dejando pocas posibilidades de expansión. Así mismo, de los tres canales de DMA, la unidad de disco y el disco duro ocupaban dos, quedando uno solo libre. La capacidad de direccionamiento de solo 1 MB y el bus de datos de *8 bits* también se constituyeron en un cuello de botella, que hacía los equipos muy lentos para las nuevas aplicaciones generadas en el desarrollo de nuevo *software.*

Con la aparición del microprocesador 80286 de 16 *bits y* los computadores AT en 1984, se diseñó en forma muy inteligente un nuevo bus, el ISA de 16 *bits,* que utilizaba en gran parte el bus anterior ISA de *8 bits* agregándole un segundo conector de 36 pines alineado con el primero con nuevas señales, como se ve en la figura 4.3.2. De esta forma, se podían utilizar la gran cantidad de tarjetas periféricas y de expansión que existían en el mercado. En esencia se agregaron otros *8 bits* de datos, más direcciones, cinco interrupciones y cuatro canales de DMA y algunas señales de control. Así mismo, se incrementó la velocidad a 8.33 MHz.

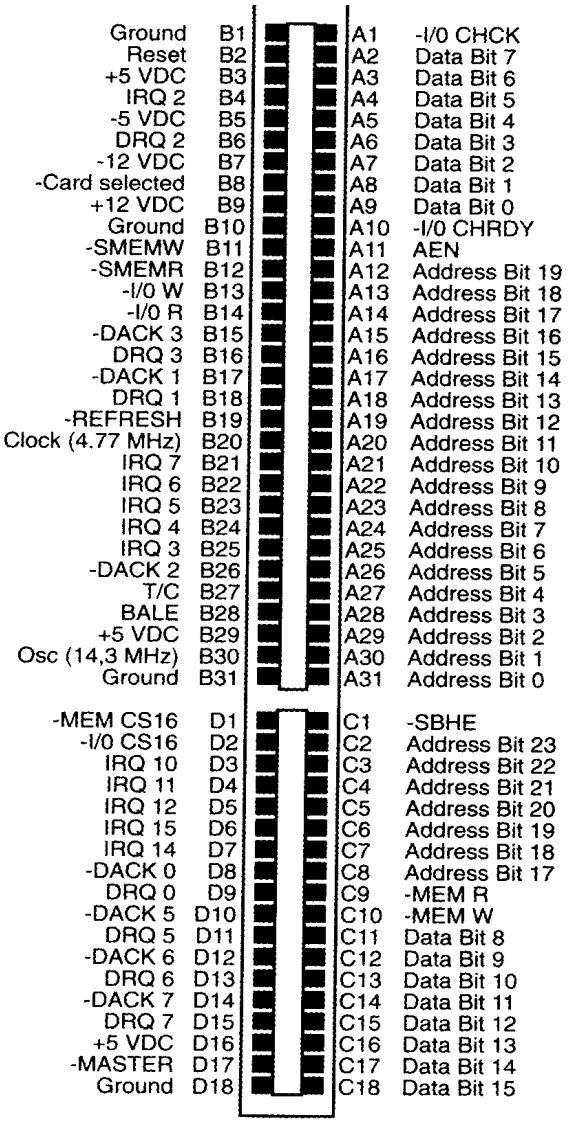


Figura 4.3.2 – Bus ISA de 16 bits

Vale la pena mencionar que el bus ISA de 16 bits o bus AT, fue ampliamente utilizado por los fabricantes de clones y compatibles, revolucionando desde el punto de vista de los precios, la industria de los PC's. Este bus todavía se encuentra en la mayoría de los computadores personales fabricados actualmente, ya que para él se fabrican una gran cantidad de tarjetas para periféricos y de expansión.

4.3.2 El bus MCA de 32 bits

Con la introducción de los microprocesadores de *32 bits* como el 80386 y el 80486, el bus ISA de 16 *bits* se quedó nuevamente atrás y se creó la necesidad de un nuevo bus. En 1987, IBM consideró que se debía archivar el bus tipo ISA y diseñó el bus MCA *(Micro Channel Architecture)* o Micro Canal en su línea de computadores PS/2.

Su diseño buscaba, además del objetivo de mejorar el rendimiento de los equipos, frenar el avance de la industria de los clones y los compatibles, ya que IBM no entregó sus características tratando de fabricar sus propias tarjetas de expansión y periféricos. Finalmente, este intento no tuvo éxito y se convirtió en un *boomerang* contra el gigante azul, que más tarde tuvo que retroceder y volver a los buses tradicionales, debido a que el bus MCA no llegó a ser muy popular entre los fabricantes de tarjetas de expansión y a que el público se negó a abandonar sus inversiones en *hardware y software* para cambiar por un nuevo sistema.

La principal diferencia entre los buses ISA y MCA consiste en la configuración física, ya que el último tiene una separación entre los pines de 0.050" con el fin de acomodar más conexiones en el mismo espacio. Esto genera una incompatibilidad total entre los dos sistemas, lo que fue un error de IBM al no considerar las inversiones en tecnología por parte de los usuarios.

4.3.3 El bus EISA de 32 bits

En 1988 y 1989 fue desarrollado el bus EISA *(Extended Industry Standard Architecture)* de *32 bits* para satisfacer las necesidades de mayor velocidad y desempeño que proporcionaban los microprocesadores 80386 y 80486 y para no dejar solo a IBM, y su bus MCA, con este mercado.

Este bus, cuyo éxito es la compatibilidad con los buses anteriores, el ISA de 8 y 16 *bits,* es un producto de un consorcio (más conocido como la banda de los nueve), formado por Compaq, Hewlett Packard, NEC, Zenith, AST, Epson, Wyse, Olivetti y Tandy. Para lograr la compatibilidad entre los dos sistemas ISA y EISA, se diseñó una solución muy ingeniosa que tardó alrededor de dos años para ser perfeccionada. E1 conector que recibe las tarjetas tiene dos filas de pines, con diferente altura o nivel y la transferencia entre los dos buses se realiza automáticamente dependiendo de la forma del conector de la tarjeta.

Los pines superiores corresponden a las tarjetas tipo ISA y en el conector inferior hay cinco topes de plástico que no permiten que éstas hagan contacto con los pines inferiores, que corresponden al bus EISA. Las tarjetas EISA tienen una serie de ranuras que coinciden con los topes y éstas se pueden deslizar hasta el fondo obteniendo el contacto con sus respectivos pines.

El bus EISA tiene 99 pines por cada lado en los cuales se ha reducido su espaciamiento de 0.100" a 0.050". En la figura 4.3.3 tenemos la distribución de las señales en este tipo de bus. Como se puede observar en la figura, el bus EISA tiene 30 líneas de direccionamiento, *32 bits* de datos, 15 niveles de interrupción y 7 canales de DMA.

Una de las principales ventajas del bus EISA es su capacidad de manejo autónomo del bus *(bus-mastering),* que en pocas palabras podría explicarse como la capacidad para permitir el intercambio de información entre dos periféricos sin la intervención de la CPU o µP Aunque la velocidad del bus EISA se conservó en 8.33 MHz, para permitir la compatibilidad con ISA, su velocidad para transferencia de los datos es mucho mayor debido al aumento en la anchura del bus; esta llega a ser de unos 33 Mb/s *(Megabits* por segundo).

Desafortunadamente, muchas tarjetas de interface tipo EISA no son más rápidas que sus similares tipo ISA, debido a que sus fabricantes no utilizan todas las propiedades que el bus ofrece. Así mismo, la baja velocidad del disco duro (comparada con otros periféricos); es un problema que no depende del bus en sí.

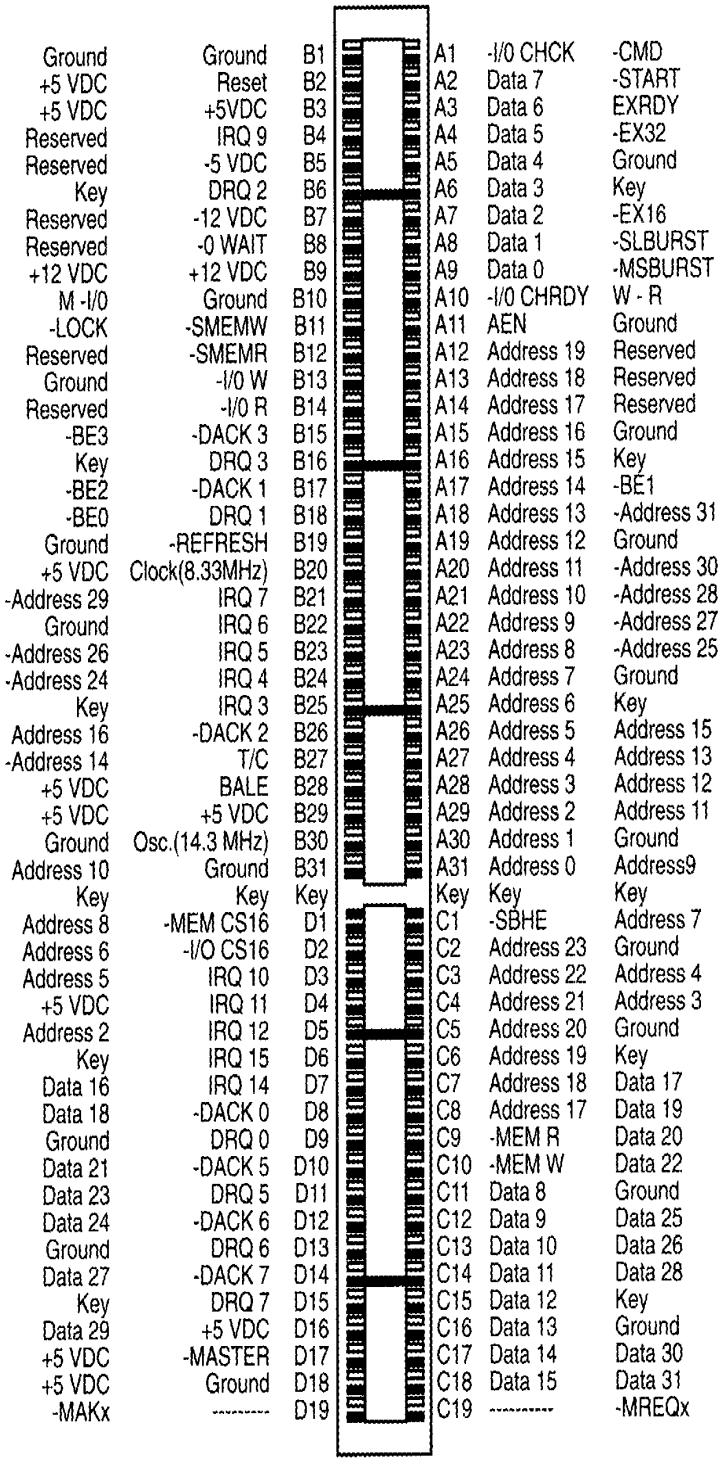


Figura 4.3.3 – Bus EISA

LOS BUSES LOCALES

Los buses locales surgieron por la constante demanda de una mayor velocidad en las operaciones que involucran gráficas, video y sonido presentes en la gran cantidad de aplicaciones con multimedia y otras similares, tan populares actualmente. La baja velocidad de los buses ISA, MCA y EISA sólo se podía superar conectándose directamente a las líneas de datos y direccionamiento del microprocesador y de la memoria. De esta forma, se pueden lograr velocidades hasta de 66 MHz en comparación a los 8 0 10 MHz de los otros buses. Los dos tipos principales de buses locales son el Video Local y el PCI.

4.3.4 El bus local de video o Video Local (VL) bus

Este tipo de bus fue propuesto en 1992 por la *Video Electronics Standards Association* y por eso se le llama *VESA Video Local Bus.* Irónicamente, este bus utiliza el mismo conector de 62 pines del bus MCA. Sus pines se muestran en la figura 4.3.4. Las tarjetas que utilizan este bus tienen dos conectores, uno en cada extremo y para cada bus. Los conectores o ranuras del *motherboard,* se colocan alineados con los conectores o ranuras ISA.EI bus VL es expandible de 32 a 64 *bits* permitiendo su utilización en sistemas con el microprocesador Pentium, por medio de otro conector tipo MCA alineado con el primero.

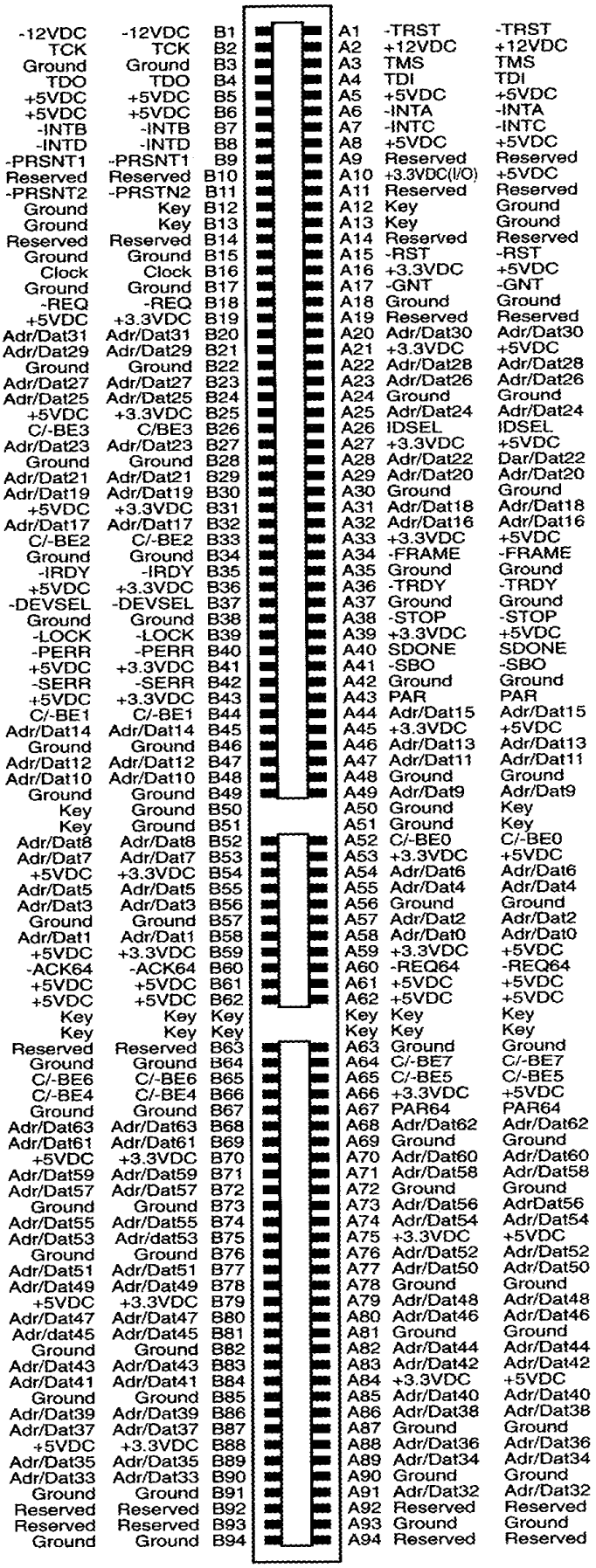


Figura 4.3.4 – Bus VESA/MCA

4.3.5 El bus local PCI

El bus local PCI *(Peripheral Component Interconnect)* fue propuesto por Intel en 1992 y apoyado por la mayoría de los fabricantes de computadores personales, incluyendo a Apple Computer, quien lo ha adoptado para su línea de equipos más reciente que utilizan el microprocesador PowerPC 604.

El bus PCI, de 188 pines, también utiliza conectores de 62 pines tipo MCA que se instalan algunas veces alineados con las ranuras ISA o EISA. En cambio del bus VL, el bus PCI utiliza una técnica de multiplexación de direcciones y datos que permite una considerable reducción en el número de pines. Esto crea la necesidad de circuitos de acoplamiento *(buffers)* que reducen un poco la velocidad de transferencia.

El bus PCI, de *32 bits,* es expandible a 64 *bits y* soporta la lógica de 3,3 voltios junto a la normal de 5 voltios. De esta forma hay cuatro versiones, la de 5 voltios y 32 *bits,* la de 3.3 voltios y 32 *bits,* la de 5 voltios y 64 *bits* y la de 3.3 voltios y 64 *bits.* En la figura 4.3.5 tenemos la disposición de los pines y el nombre de las diferentes señales.

Las ranuras PCI estarán dedicadas para los periféricos más críticos que pueden estar conectados directamente a la tarjeta principal o por medio de tarjetas de interface como son las unidades para multimedia (CD-ROM), tarjetas para gráficas y video, unidades de disco duro y tarjetas para conexión en red.

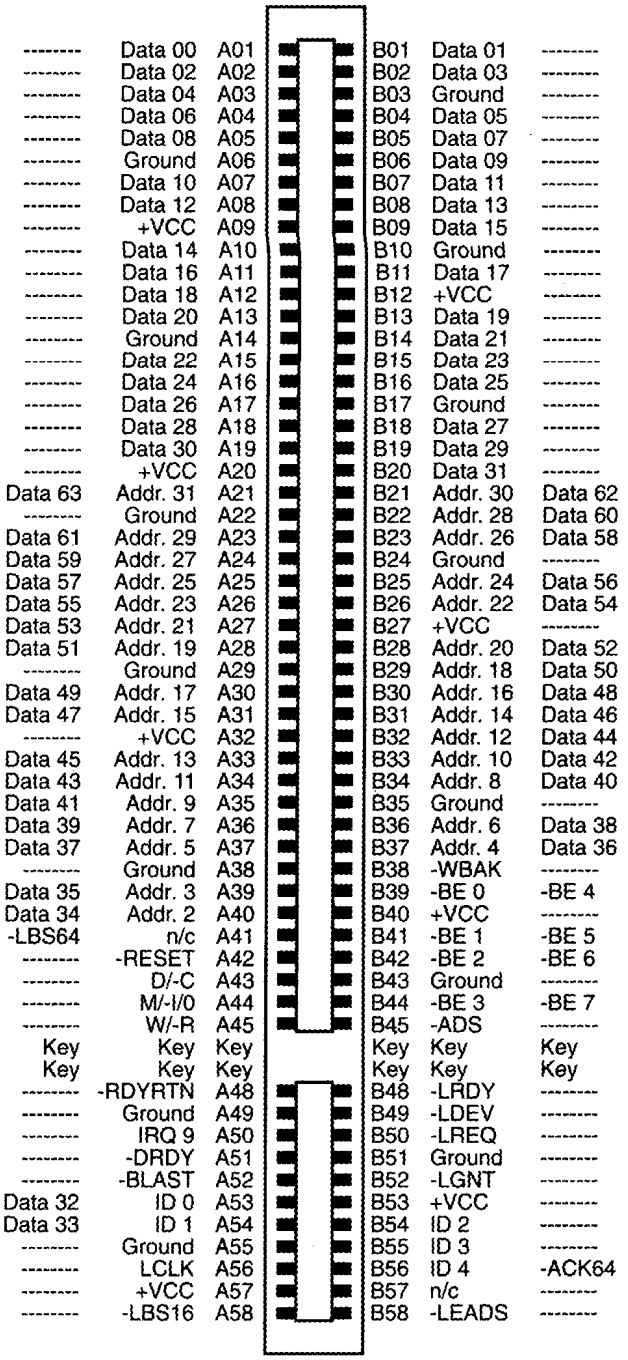


Figura 4.3.5 – Bus PCI

Cada paso que se avanza en la industria de los computadores personales trae cambios tanto técnicos como de adaptación del usuario y requiere un consenso de las empresas que dominan el mercado. Si no se tiene en cuenta al usuario o no se logra una norma aprobada por un buen número de fabricantes, como en el caso del bus MCA, los resultados pueden ser muy negativos. El bus PCI, aceptado por las dos principales plataformas, PC y Mac, y en versiones de 32 y 64 bits, es uno de los que permaneció por más tiempo en el mercado.

4.3.6 El Puerto de Gráficos Acelerado (AGP)

La necesidad de aumentar el ancho de banda (ancho de banda) entre el procesador principal y el subsistema video lleva originalmente al desarrollo del bus de I/O local en el PCs, empezando con el VESA local bus y luego llevado al popular bus PCI. Esta tendencia continua de la necesidad de ancho de banda de video comenzó a empujar contra los límites del bus PCI.

Tal como anteriormente fue el caso con el bus ISA, el tráfico en el bus PCI está empezando a ponerse pesado en las PCs de extremo alto, con video, disco duro y datos de periféricos todos que compiten para el mismo ancho de banda de I/O. Para combatir la saturación eventual del bus PCI con información de video, una nueva interface ha sido introducida por Intel, diseñado específicamente para el subsistema de video. Se llama el Puerto de Gráficos Acelerado o AGP.

AGP se desarrolló en respuesta a la tendencia hacia los requisitos de mayores y mayores desempeño para el video. Cuando el software evoluciona y el uso de la computadora continúa en las áreas previamente inexploradas como aceleración 3D y reproducción de video en pleno movimiento, el procesador y los procesadores gráficos necesitan procesar cada vez más información. El bus de PCI alcanzó su límite de desempeño en estas aplicaciones, sobre todo con disco duro y también otros periféricos en su lucha por el mismo ancho de banda.

Otro problema ha sido las demandas crecientes para la memoria de video. Cuando la informática en 3D se vuelve más habitual, se requieren cantidades muy grandes de memoria, no sólo para la imagen de la pantalla sino también para hacer los cálculos en 3D. Esto tradicionalmente ha significado poner más memoria en la tarjeta de video para hacer este trabajo. Hay dos problemas con esto:

Costo: La memoria de la tarjeta de video es muy cara comparada a la RAM normal del sistema.

Tamaño limitado: La cantidad de memoria en la tarjeta de video estaba limitada: si usted decide poner 6 MB en la tarjeta y se necesitan 4 MB para el buffer de marcos, usted tiene sólo 2 MB para trabajo de procesamiento. No es fácil de extender esta memoria, y usted no puede usarla para nada más que no sea para el proceso de video.

AGP soluciona estos problemas permitiendo al procesador de video acceder a la memoria del sistema principal para hacer sus cálculos. Esto es más eficaz porque esta memoria puede compartirse dinámicamente entre el procesador del sistema y el procesador de video, dependiendo de las necesidades del sistema.

La idea detrás de AGP es simple: crear una interfase más rápida y especializada, entre el procesador de video y el procesador del sistema. La interfase es sólo entre estos dos dispositivos; esto tiene tres ventajas principales: Es más fácil de implementar el puerto, hace más fácil aumentar la velocidad del AGP, y hace posible implementar mejoras en el diseño que son específicas del video.

AGP es considerado un puerto, y no un bus, porque sólo involucra dos dispositivos (el procesador y tarjeta del video) y no es expansible. Una de las grandes ventajas de AGP es que aísla el subsistema de video del resto de la PC de manera que no hay casi ninguna disputa sobre el ancho de banda de I/O como sucedía con el PCI. Con la tarjeta de video quitada del bus de PCI, otros dispositivos de PCI también se beneficiarán del ancho de banda mejorado.

Interfaz AGP

La interfaz AGP todavía es bastante similar a PCI. El propio *slot* es físicamente similar en forma y tamaño, pero se desplaza más allá del borde del motherboard en donde se ubican los *slots* PCI. La especificación de AGP es de hecho basada en la especificación PCI 2.1 que incluye

un alto ancho de banda, de 66 MHz que nunca antes fue implementado en el PC. Los *motherboards* con AGP tienen un solo *slot* de tarjeta de expansión para la tarjeta de video AGP, y normalmente un *slot* PCI menos, y es por otra parte bastante similar a las *motherboards* PCI.

AGP tiene de 32 bits de ancho, igual que el PCI, pero en lugar de correr a la mitad de la velocidad del bus del sistema (memoria) de la manera que lo hace PCI, corre a velocidad plena del bus. Esto significa que en un motherboard Pentium II normale, AGP usa 66 MHz en lugar de los 33 MHz del bus PCI. Esto duplica el ancho de banda del puerto; en lugar del límite de 133 MB/s de PCI, AGP, en su modo de velocidad más bajo tiene un ancho de banda de 266 MB/s. Además, por supuesto, los beneficios de no tener que compartir ancho de banda con otros dispositivos de PCI.

Además de duplicar la velocidad del bus, AGP ha definido un modo 2X que usa señalización especial para permitir enviar a través del puerto el doble de datos a la misma velocidad de reloj. Lo que el hardware hace es enviar información sobre los flancos de subida y bajada de la señal de reloj. El resultado es que el desempeño se duplica de nuevo, a 533 MB/s que el ancho de banda teórico. En el modo 4X la tasa de transferencia se eleva a 1066 MB/s y en el modo 8X se logra un ancho de banda de 2.132 GB/s. Este bus ha sido reemplazado en la actualidad por el PCI-E.

4.3.7 Interfaz PCI Express (PCI-E)

A diferencia de todos los anteriores, PCI Express está implementado con conexiones serie de alta velocidad, punto a punto y bidireccionales, llamadas “lanes”. En la figura siguiente se puede ver un “lane”, correspondiente a PCI-E 1X.

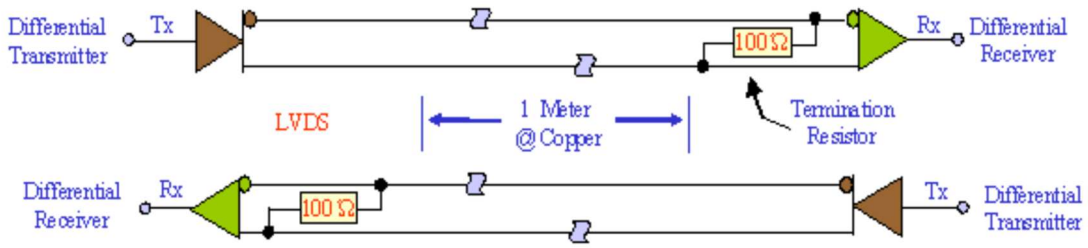


Figura – 5.3.6 – circuito de la interfaz PCI-E

La velocidad de la señal se estableció en 2,5 GHz y los datos se transmiten en “frames” de 10 bits, con uno de “start”, 8 de datos y uno de “stop”. De esta forma, a 2,5 GHz se pueden transmitir 250 MB/s. Esto requiere de menor cantidad de contactos que una interfaz paralelo, por lo que el conector será más pequeño. Una revisión posterior del estándar aumentó la velocidad a 5 GHz, con lo que el máximo ancho de banda, usando 23 “lanes” se eleva a 16 GB/s.

En las figuras siguientes se puede ver la lista de señales y las dimensiones de los conectores.

Pin	Side B Connector		Side A Connector	
#	Name	Description	Name	Description
1	+12v	+12 volt power	PRSNT#1	Hot plug presence detect
2	+12v	+12 volt power	+12v	+12 volt power

3	RSVD	Reserved	+12v	+12 volt power
4	GND	Ground	GND	Ground
5	SMCLK	SMBus clock	JTAG2	TCK
6	SMDAT	SMBus data	JTAG3	TDI
7	GND	Ground	JTAG4	TDO
8	+3.3v	+3.3 volt power	JTAG5	TMS
9	JTAG1	+TRST#	+3.3v	+3.3 volt power
10	3.3Vaux	3.3v volt power	+3.3v	+3.3 volt power
11	WAKE#	Link Reactivation	PWRGD	Power Good
Mechanical Key				
12	RSVD	Reserved	GND	Ground
13	GND	Ground	REFCLK+	Reference Clock Differential pair
14	HSOp(0)	Transmitter Lane 0, Differential pair	REFCLK-	
15	HSOn(0)		GND	Ground
16	GND	Ground	HSIp(0)	Receiver Lane 0, Differential pair
17	PRSNT#2	Hotplug detect	HSIn(0)	
18	GND	Ground	GND	Ground

Tabla 4.3.1 – Lista de señales de PCI-E 1X

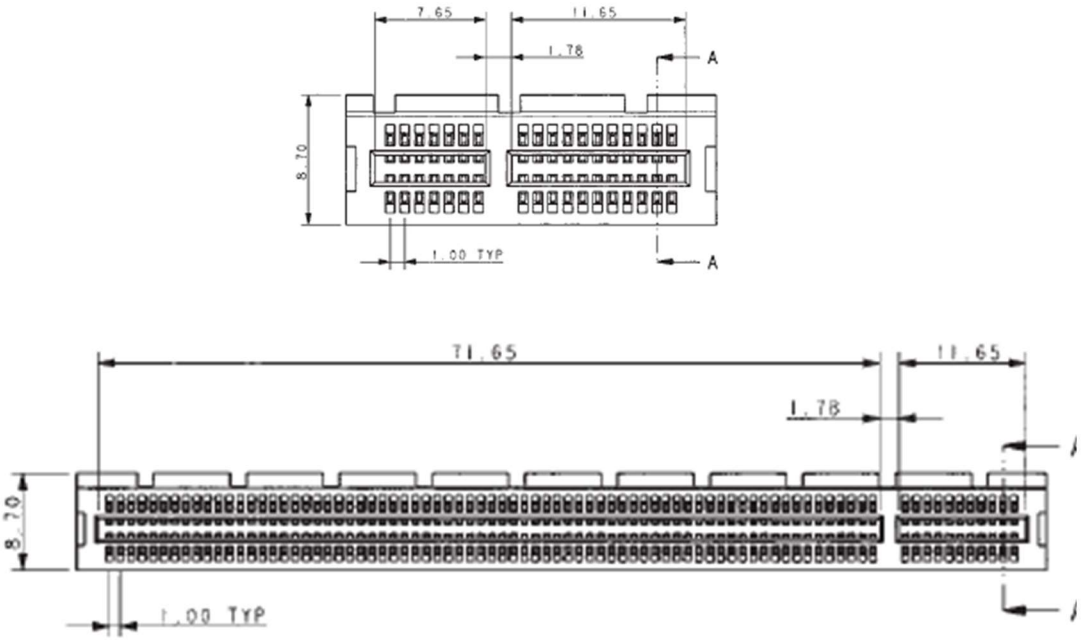


Figura – 5.3.7 – Conectores PCI-E 1X y 16X

4.3.8 El SMBus

Introducción

El bus de Dirección de Sistema (System Management Bus - SMBus) es una interfaz de dos conductores a través de la cual varios componentes (chips) de sistema pueden comunicarse entre sí y con el resto del sistema. Está basado en los principios de funcionamiento del I^2C *.

SMBus provee un bus de control para tareas del sistema y manejo de potencia. Un sistema puede usar SMBus para pasar mensajes hacia y desde los dispositivos en lugar de utilizar líneas del comando individuales. Quitando las líneas del comando individuales se reduce la cantidad de pines. La aceptación de mensajes asegura la expansibilidad futura.

Con el Bus de Dirección de Sistema, un dispositivo puede proporcionar la información del fabricante, decir al sistema su número de parte o modelo, salvar su estado para un evento suspendido, informar los diferentes tipos de errores, aceptar los parámetros de control y devolver su estado (status).

Características generales

SMBus es un bus de dos conductores. Múltiples dispositivos, tanto los amos del bus como los esclavos del bus, pueden conectarse a un segmento de SMBus. Generalmente, un dispositivo amo de bus comienza una transferencia de bus entre él y un solo esclavo del bus y proporciona las señales de reloj. La única excepción a esta regla, detallada después, es durante el arreglo inicial de bus (setup) cuando un solo amo puede comenzar las transacciones con múltiples esclavos simultáneamente. Un dispositivo esclavo de bus puede recibir datos proporcionados por el amo o puede proporcionar datos al amo.

Sólo un dispositivo puede dominar el bus en un momento dado. Debido a que más de un dispositivo puede intentar tomar mando del bus como amo, SMBus proporciona un mecanismo del arbitraje basado en la conexión AND-cableado de todas las interfases de dispositivos unidos al SMBus.

Esta especificación define dos clases de características eléctricas, baja potencia y alta potencia. La primera clase, originalmente definida en las especificaciones SMBus 1.0 y 1.1, se diseñó principalmente pensando en las Baterías Inteligentes, pero podría usarse con otros dispositivos de baja potencia. Esta versión de la especificación introduce un conjunto alternativo de características eléctricas de alta potencia. Esta clase es apropiada para el uso cuando se requiere capacidad de manejo más alta, por ejemplo con los dispositivos SMBus incluídos en tarjetas PCI y para conectar estas tarjetas a otras a través del conector PCI entre sí y a los dispositivos SMBus en la placa madre del sistema.

Los dispositivos pueden ser alimentados por VDD en el bus o por otra fuente de poder, VBus, (como, por ejemplo, las Baterías Inteligentes) e inter-operarán con tal de que adhieran a las especificaciones eléctricas SMBus para su clase.

El diagrama siguiente muestra una aplicación de ejemplo de un SMBus 5 voltios con dispositivos alimentados por el bus VDD inter-operando con dispositivos auto-alimentados.

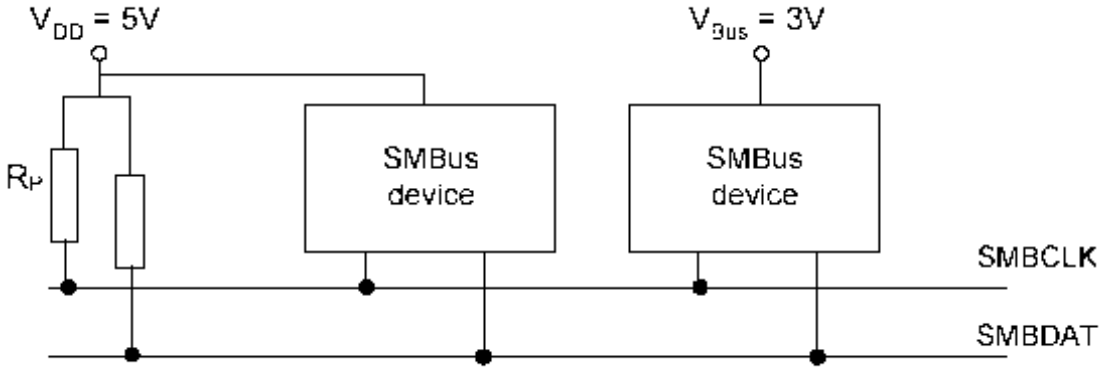


Figura 4.3.8 – Aplicación de SM-Bus

VDD puede ser 3 a 5 voltios + / - 10% y puede haber dispositivos de SMBus alimentados directamente por el bus VDD. Ambas líneas, SMBCLK y SMBDAT son bi-direccionales, conectadas a un voltaje de alimentación positivo a través de una resistencia de pull-up, una fuente de corriente u otro circuito similar. Cuando el bus esta libre, ambas líneas están altas. Las etapas de salida de los dispositivos conectados al bus deben tener un drenaje abierto o colector abierto para realizar la función AND-cableada.

Capa 1 - Capa Física

Se espera que SMBus 2.0 opere en por lo menos dos ambientes mutuamente exclusivos diferentes que tienen requisitos eléctricos diferentes. En un caso, SMBus debe operar fiablemente en el ambiente de baja potencia tradicional de las baterías. También debe operar fiablemente en el ambiente de mayor ruido del conector de PCI y las tarjetas PCI.

Las especificaciones de CA son comunes, tanto para el de baja como de alta potencia. Las mismas se muestran en el diagrama siguiente:

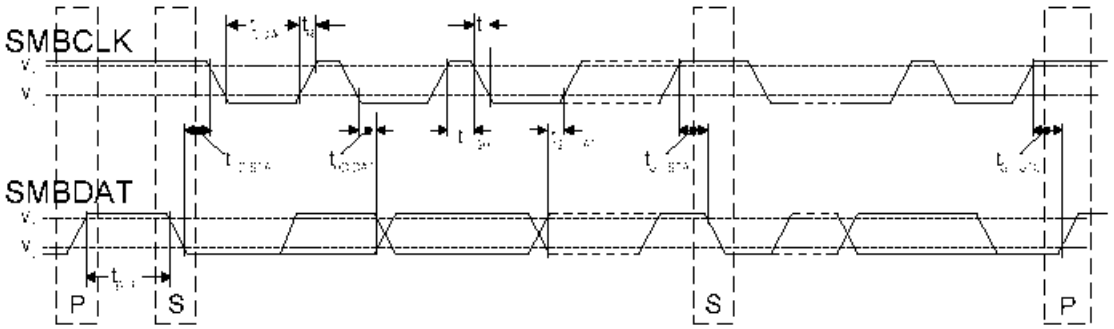


Figura 4.3.9 – Diagrama de tiempos de SM-Bus

Capa 2 – Capa de Enlace de Datos

SMBus usa niveles de voltaje fijos para definir los estados lógicos CERO y UNO en el bus.

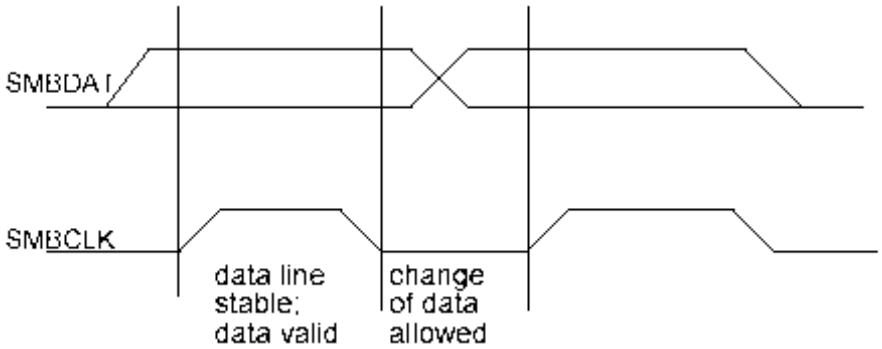


Figura 4.3.10 – Diagrama de tiempos de SM-Bus

Cada byte transferido en el bus debe ser seguido por un bit de reconocimiento (ACK). Los bytes se transfieren con el bit más significativo (MSB) primero.

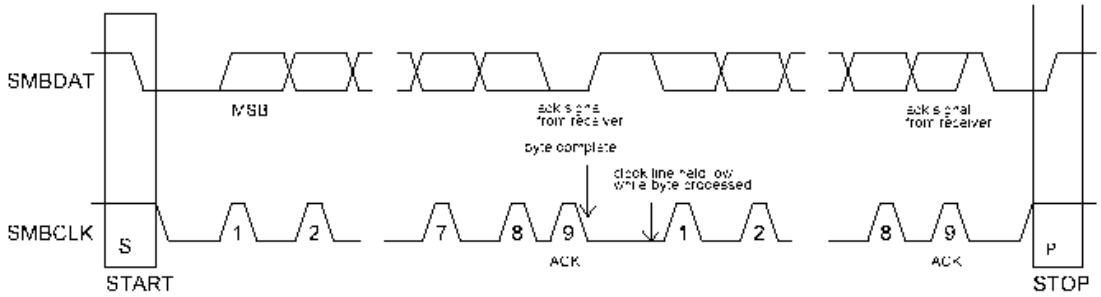


Figura 4.3.11 – Protocolo de SM-Bus

Capa 3 - Capa de Red

La Especificación de SMBus se refiere a tres tipos de dispositivos. Un esclavo (slave) es un dispositivo que está recibiendo o está respondiendo a un orden. Un amo (master) es un dispositivo que emite los órdenes, genera los relojes y termina la transferencia. Un organizador (host) es un amo especializado que proporciona la interfaz principal al CPU del sistema. Un organizador debe ser un amo-esclavo y debe soportar los protocolos de notificación de host de SMBus. Puede haber a lo sumo un organizador en un sistema. Un ejemplo de sistema "hostless" es un simple estación de carga de baterías. La estación podría estar enchufada en la pared esperando para cargar una batería inteligente.

Referencias:

System Management Bus (SMBus) Specification Version 2.0 www.smbus.org

4.3.9 Resumen

Mientras que la performance de los procesadores continúa duplicándose cada 18 meses, la performance de los buses de IO se retrasa, doblando la performance cada tres años, como se aprecia en la figura 4.3.12.

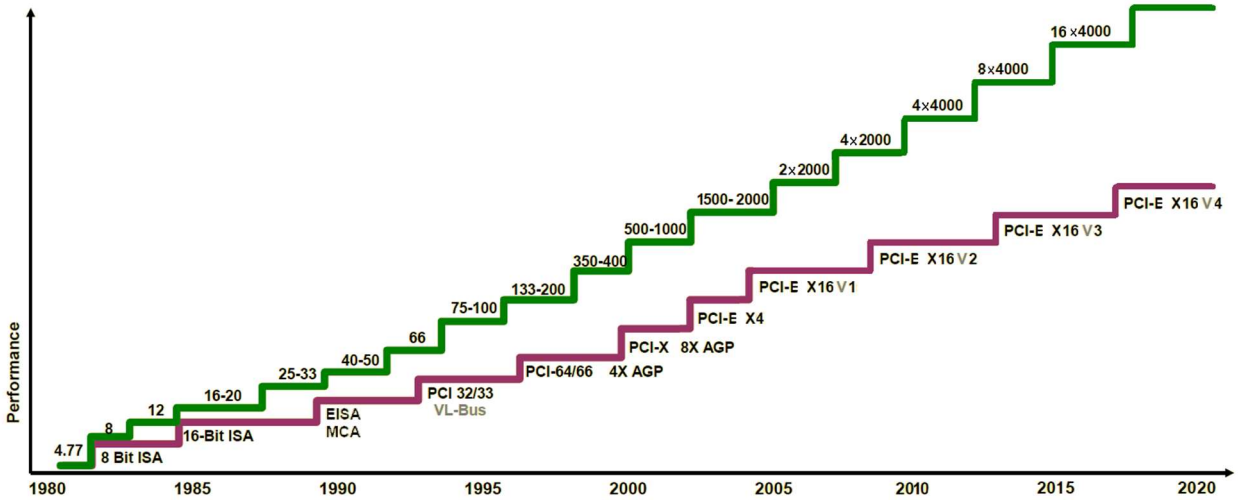


Figura 4.3.12 – Performance de procesadores vs Buses

Las características de los diferentes buses son las siguientes:

Interfaz	Tensión	Frecuencia	Datos por ciclo	Líneas de datos	Ancho de banda
ISA-8	5 V	8 MHz	1	8	**8 MB/s**
ISA-16	5 V	8 MHz	1	16	**16 MB/s**
VLB	5 V	33 MHz	1	32	**133 MB/s**
PCI 2.0	5 V	33 MHz	1	32	**133 MB/s**
PCI 3.0	3.3 V	33 MHz	1	32	**133 MB/s**
PCI-X 2.0	3.3 V	266 MHz	1	64	**2133 MB/s**
AGP 1.0	3,3 V	66 MHz	1 ó 2	32	**266/533 MB/s**
AGP 2.0	1,5 V	66 MHz	4	32	**1066 MB/s**
AGP 3.0	0,9 V	66 MHz	8	32	**2133 MB/s**
PCI-E 1.0 X1	3,3 V	2500 MHz	1	1	**250 MB/s**
PCI-E 2.0 X1	3,3 V	5000 MHz	1	1	**500 MB/s**
PCI-E 2.0 X4	3,3 V	5000 MHz	1	4	**2000 MB/s**
PCI-E 2.0 X16	3,3 V	5000 MHz	1	16	**8000 MB/s**
PCI-E 3.0 X16	3,3 V	8000 MHz	1	16	**12800 MB/s**
PCI-E 4.0 X16	3,3 V	16000 MHz	1	16	**25600 MB/s**

Tabla 4.3.1 – Características de los buses

4.4 ARQUITECTURAS ACTUALES DE PC

4.4.1 Introducción:

En el anexo A.1 se pueden ver algunos modelos de motherboards, desde el primero, diseñado para el 8088 (o su clon, el NEC V20) hasta uno actual (Diciembre de 2007), que permite instalar los últimos procesadores de 4 núcleos, dos placas de video en SLI, varios GB de RAM DDR2, arreglos de discos SATA2, audio de 8 canales, Ethernet de 1 GB, Firewire, varios puertos USB 2.0 y una Bios que permite múltiples configuraciones del HW sin abrir el gabinete.

4.4.2 Moterboard Intel

En el diagrama siguiente se puede ver la interconexión entre los distintos bloques que constituyen una motherboard actual para procesadores Intel.

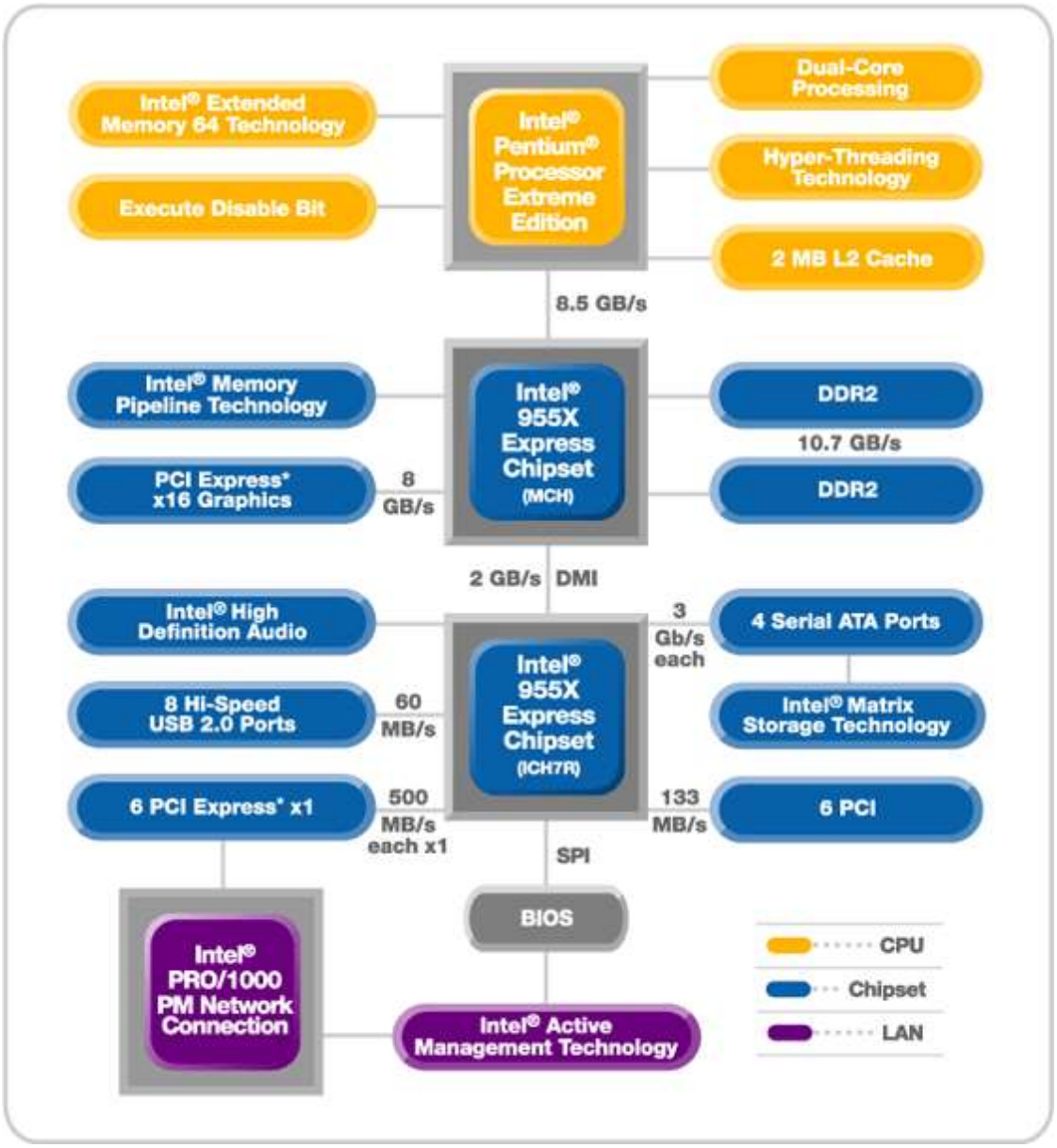


Figura 4.4.1 – Diagrama en bloque de motherboard para Intel

4.4.3 Moterboard AMD

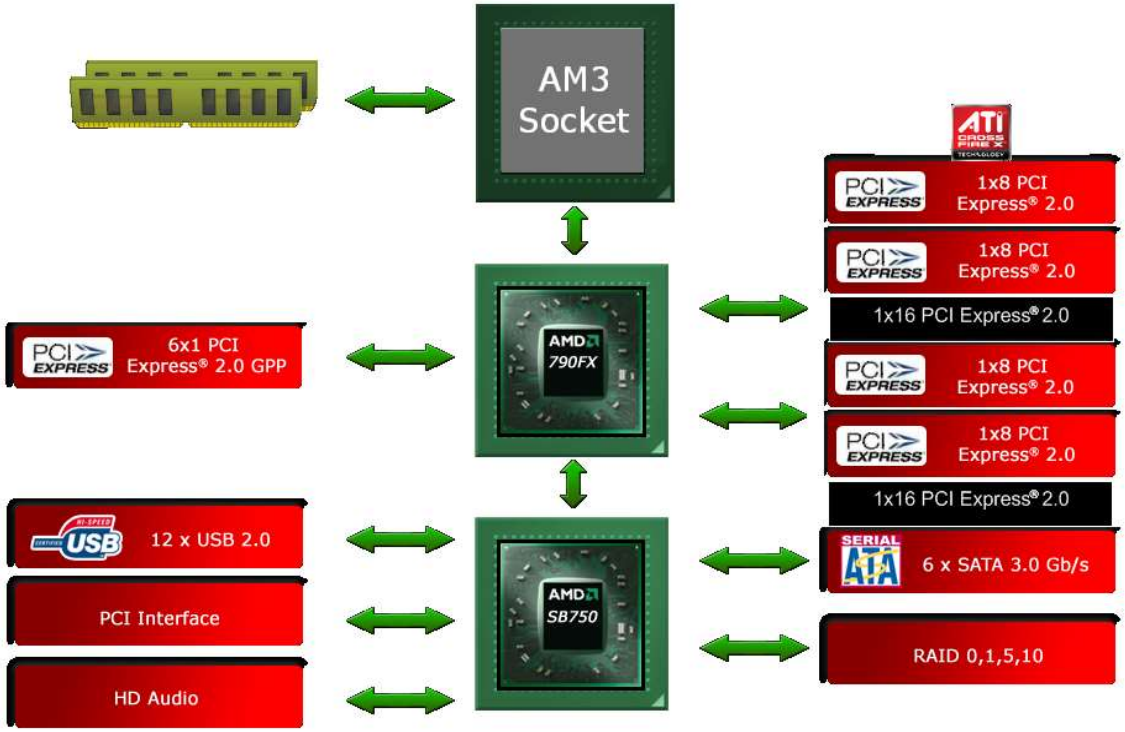


Figura 4.4.2 – Diagrama en bloque de motherboard para AMD

4.4.4 BIOS Setup de un motherboard actual

Figura 4.4.3 – Interfaz UEFI (Unified Extensible Firmware Interface)

4.5 MONITORES DE VIDEO

4.5.1 Fundamentos

Un monitor es un dispositivo de proyección de imágenes en la pantalla. Su calidad es determinada por su habilidad de producir la mejor imagen posible. Esto significa buenos colores e imágenes y textos claros y nítidos.

4.5.2 Colores

Los colores son rayos de luz, es decir ondas electromagnéticas con longitudes de onda entre 380 nm y 780 nm. Nosotros los percibimos con nuestros ojos y nuestro cerebro los traduce en lo que nosotros llamamos "colores".

En otros términos: los colores son productos de nuestro cerebro. Esto significa que una persona puede concebir los colores de manera diferente de otra.

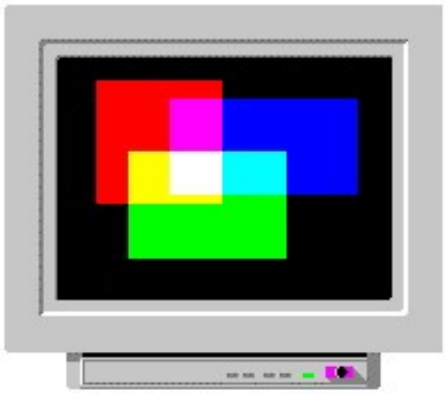


Figura 4.5.1 – Monitor color

Para mostrar los colores, los monitores usan lo que se llama **"mezcla de colores aditiva"**, usando luz verde, roja y azul. Esto es debido al hecho que si nosotros mezclamos luz roja, verde y azul (RGB), conseguimos luz blanca. Cuando se requiere blanco en la pantalla, se encienden los puntos rojos, verdes y azules que al brillar juntos producen la luz blanca.

Profundidad de color (Color depth)

En la tabla siguiente se muestran las diferentes cantidades de bits asignadas por pixel, lo que determina la cantidad de colores diferentes que podrá mostrar el monitor, dependiendo de sus posibilidades y de la tarjeta gráfica. Programas de edición pueden usar hasta 48 bits.

	Otro uso	Rojo (R)	Verde (G)	Azul (B)	Total/pixel	Colores
	1 bit	1 bit	1 bit	1 bit	4 bits	16
	0 bits	3 bits	3 bits	2 bits	8 bits	256
Color alto	1 bit	5 bits	5 bits	5 bits	16 bits	65,535
Color verdadero	0 bits	8 bits	8 bits	8 bits	24 bits	16,777,216
Color profundo	2 bits	10 bits	10 bits	10 bits	32 bits	1,073,741,824

Tabla 4.5.1 – Bits por pixel y cantidad de colores

Comparativa de la misma imagen, mostrada con diferentes profundidades de color.

24 bit.png, 16,777,216 colores, 98 kB

8 bit.png, 256 colores, 37 kB (−62%)

4 bit.png, 16 colores, 13 kB (−87%)

Figuras 4.5.2 a, b y c – Profundidad de color

Cuando se trata de representar colores en un papel, por medio de tintas o en otras superficies mediante pinturas, se utiliza la **"mezcla de colores sustractiva",** usando los colores cyan, magenta y amarillo (CMY) como los empleados en la tinta de las impresoras. Las impresoras usualmente cuentan también con un depósito de tinta negra para lograr mejor color y definición en los textos, como es el caso de este apunte.

cyan = (verde + azul) y es el complemento del rojo

magenta = (rojo + azul) y es el complemento del verde

amarillo = (rojo + verde) y es el complemento del azul

4.5.3 Ajustes en los monitores

Color

Aumentando o disminuyendo el ajuste de color afectará la saturación de color de su monitor, significando cuán profundos o realzados son los colores. Simplemente ajústelo al máximo para ver lo que es "demasiado color".

Tinte (tint)

El ajuste de tinte es uno de los ajustes más flexibles en lo que se refiere a su definición, y lo que realmente hace en su monitor puede variar dependiendo del fabricante. A veces parece como un ajuste de brillo más temperamental. En otros monitores, puede afectar tanto el color como la atenuación. Generalmente estamos hablando sobre el color "colorido" (hue), y usted sabrá cuando hace lo correcto cuando la imagen es tan real como sea posible.

Brillo (brightness)

Cambiándolo hará su pantalla más oscura o más luminosa y realmente es una cuestión de preferencia personal. La mayoría de las personas tiende a encontrar confortable aproximadamente el 75% del rango. Este ajuste puede depender de la iluminación ambiente y de hecho hay monitores que poseen un sensor de luz para realizar un ajuste automático del brillo.

Agudeza (sharpness)

Hay un poco de vaguedad sobre cuanto los ajustes de su monitor pueden ayudar a definir su agudeza realmente. Depende mas de su resolución y la calidad del material que está viendo (ej. una película en alta definición vs. un video de YouTube, o una fotografía profesional vs. una casera). Generalmente esto hará los bordes de su imagen más oscuros y más definidos. Ajustándolo demasiado bajo, resultará una imagen suave, como con niebla.

4.5.4 Tecnologías de monitores

Hay varios tipos de monitores. Los de tecnología de tubo de rayos catódicos (CRT) fueron los primeros y actualmente han sido reemplazados por los de tecnología de cristal líquido (LCD) y LED, con menor consumo y tamaño.

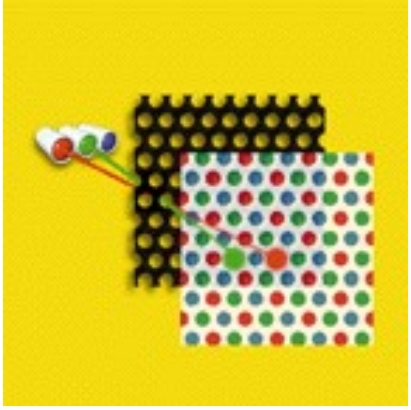
EL CRT DE TRÍADAS DE PUNTOS Y MÁSCARA DE SOMBRA

Los monitores trabajan apuntando un haz de electrones a una gota de fósforo, que a su vez brilla. Esta luz es lo que nosotros percibimos como un "pixel" (PIcture ELement - elemento de imagen) en la pantalla. El monitor a color normal tiene tres puntos (tríada de puntos) por cada posición en la pantalla: rojo, verde y azul. Hay un cañón de electrones correspondiente para cada color que emite un haz de electrones de intensidad variable - esto es para controlar el brillo. Para

asegurar que los electrones de cada cañón acierten en el fósforo correspondiente, se usa una "máscara de sombra".

Debido a que los tres haces de electrones llegan con ángulos ligeramente diferentes (de los tres cañones de electrones), es posible construir y alinear la máscara de sombra tal que el haz de electrones de un cañón golpeará el punto del fósforo correcto, pero los otros dos fósforos estarán en la sombra.

De esta manera, la intensidad de rojo, verde y azul puede ser controlada separadamente para cada posición de las tríadas de puntos. Después de que un haz deja de incidir en un punto de fósforo, el fósforo continúa brillando brevemente, esta condición se llama persistencia. Para que una imagen permanezca estable, los fósforos deben reactivarse con barridos repetidos de los haces de electrones.



Figuras 4.5.3 CRT de tríadas y mascara de sombra

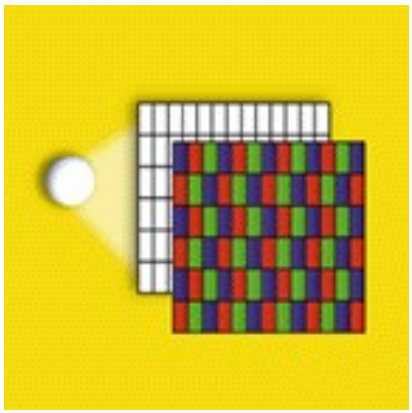
La máscara de sombra normalmente es una máscara de invar (64% hierro & 36% níquel) consistente en una placa delgada con agujeros pequeños perforados en ella. Sólo aproximadamente 20-30% del haz de electrones pasa realmente a través de los agujeros en la máscara y golpea el fósforo de la pantalla, de tal forma que el resto de la energía es disipado como calor en la máscara.

Como resultado, los monitores de máscara de sombra son propensos a tener problemas de pureza de color cuando ellos se calientan debido al pequeño cambio en la posición de los agujeros con respecto a los puntos del fósforo. Las máscaras de sombra - o su equivalente - han hecho posible la producción de masa de CRTs.

Debido a la forma en que los puntos de fósforo se colocan en los monitores de máscara de sombra, estos tubos entregan los bordes limpios y diagonales afiladas, lo que es importante para la reproducción de texto.

EL PANEL DE LCD

En un panel de LCD, una capa delgada de material transmite o bloquea la luz; con este material colocado en una serie de células diminutas, cada píxel en una imagen puede encenderse o apagarse como sea requerido. Los LCDs entregan imágenes sólidas, sin parpadeo y emiten mucho más bajas cantidades de radiación VLF (Frecuencia Muy Baja) y ELF (Frecuencia Sumamente Baja). También ocupan mucho menos lugar.



Figuras 4.5.4 Panel LCD

Historia del TFT LCD

Los cristales líquidos fueron descubiertos por el botánico austríaco Fredreich Rheinizer en 1888. A mediados de 1960, los científicos descubrieron que los cristales líquidos, cuando eran estimulados con una carga eléctrica externa, podían alterar las propiedades del pasaje de la luz a través de ellos.

El cristal TFT (Thin Film Transistor Liquid Crystal Display) tiene tantos TFTs como pixeles a mostrar, mientras que un cristal filtro de color (CFG) es el que genera el color. Los cristales líquidos se mueven de acuerdo al potencial aplicado entre el cristal CFG y el cristal TFT. La cantidad de luz suministrada por la luz posterior (Back Light) está determinada por la cantidad de movimiento de los cristales líquidos de forma de generar color.

Los monitores actuales están construidos con una estructura "sándwich", como muestra la figura:

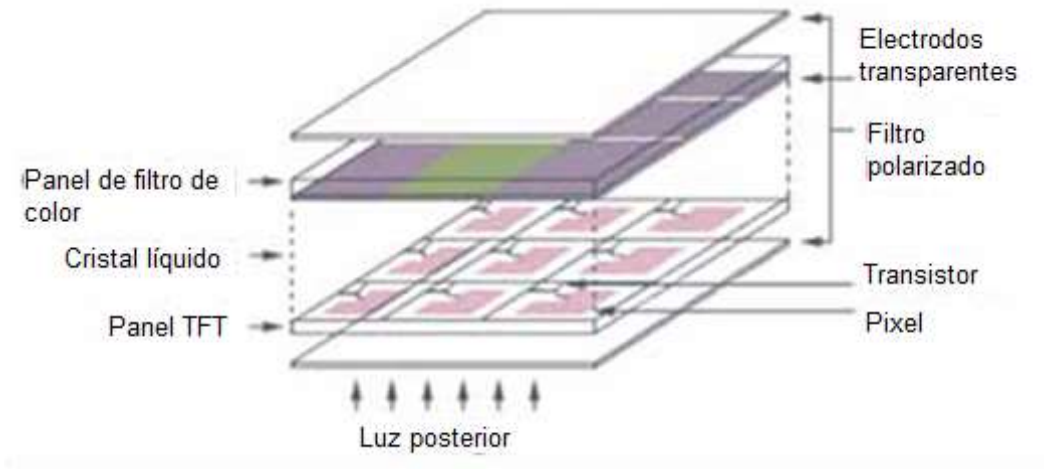


Figura 4.5.5 – Panel LCD TFT

LCD de matriz pasiva y activa

En LCDs de matriz pasiva no hay dispositivos de conmutación y cada pixel es direccionado más de una vez en una trama de tiempo. La tensión efectiva aplicada al cristal

líquido debe promediar los pulsos de tensión aplicados en las diferentes tramas de tiempo, lo que resulta en un bajo tiempo de respuesta, mayor de 150 ms y una reducción de la relación de contraste máxima.

En los LCDs de matriz activa, por el contrario, un dispositivo de conmutación y un capacitor de almacenamiento están integrados en cada punto de cruce de los electrodos.

Lo habitual es que la capa semiconductora esté hecha de silicio amorfo (a-Si).

Una tecnología alternativa, el silicio policristalino (p-Si), es costosa para producir, especialmente para pantallas grandes. Casi todos los TFT LCDs están hechos de a-Si debido a que es una tecnología madura y económica, pero la movilidad de los electrones del p-Si es uno o dos órdenes de magnitud mayor que la del a-Si, lo que lo hace especialmente apropiado para dispositivos como visores o proyectores.

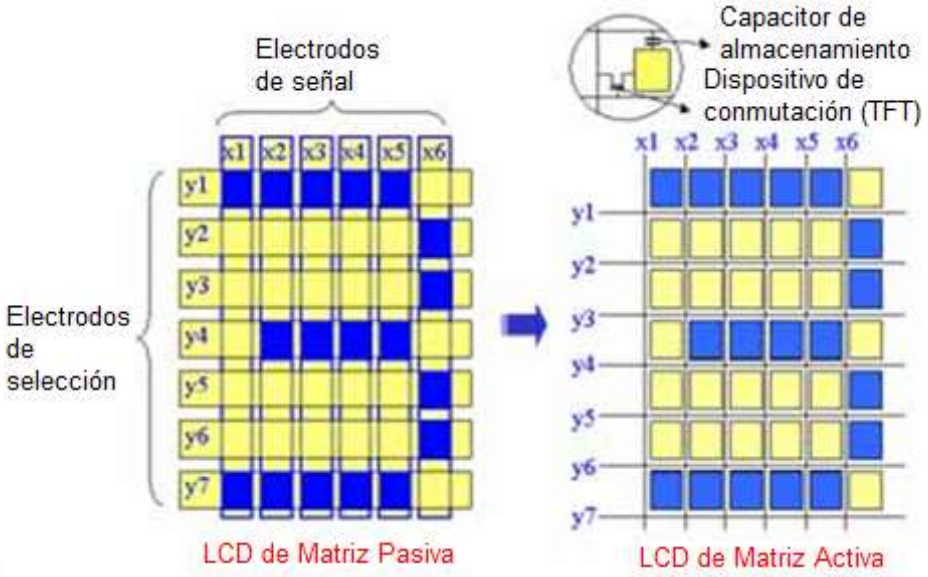


Figura 4.5.6 – Matriz pasiva y matriz activa

Estructura de un panel LCD TFT

Un panel LCD TFT contiene un número específico de pixeles unitarios, a menudo llamados subpixeles. Cada píxel unitario tiene un TFT, un electrodo de píxel (IT0) y un capacitor de almacenamiento. Por ejemplo: un panel LCD SVGA tiene un total de 800 x 3 x 600 = 1440000 pixeles unitarios.

Cada píxel unitario es conectado a una de las líneas de habilitación de bus y a una de las líneas de datos de bus en una matriz de 2400 x 600 en el caso de SVGA.

La performance de un LCD TFT está relacionada con los parámetros de diseño de los píxel unitarios, el ancho de canal y la longitud de canal del TFT, el solapamiento de los electrodos del TFT, el tamaño del capacitor de almacenamiento y el electrodo de píxel y el espaciado entre estos elementos.

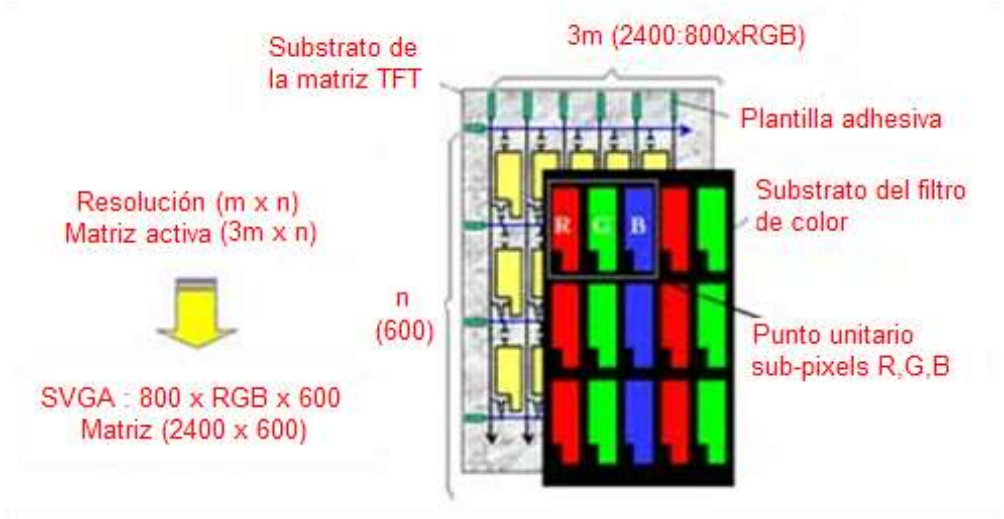


Figura 4.5.7 – Detalle de la estructura de un panel LCD TFT

Generación de color

El filtro de color de un LCD TFT consiste de tres colores primarios rojo (R), verde (G) y azul (B), que están incluidos en el substrato del filtro de colores.

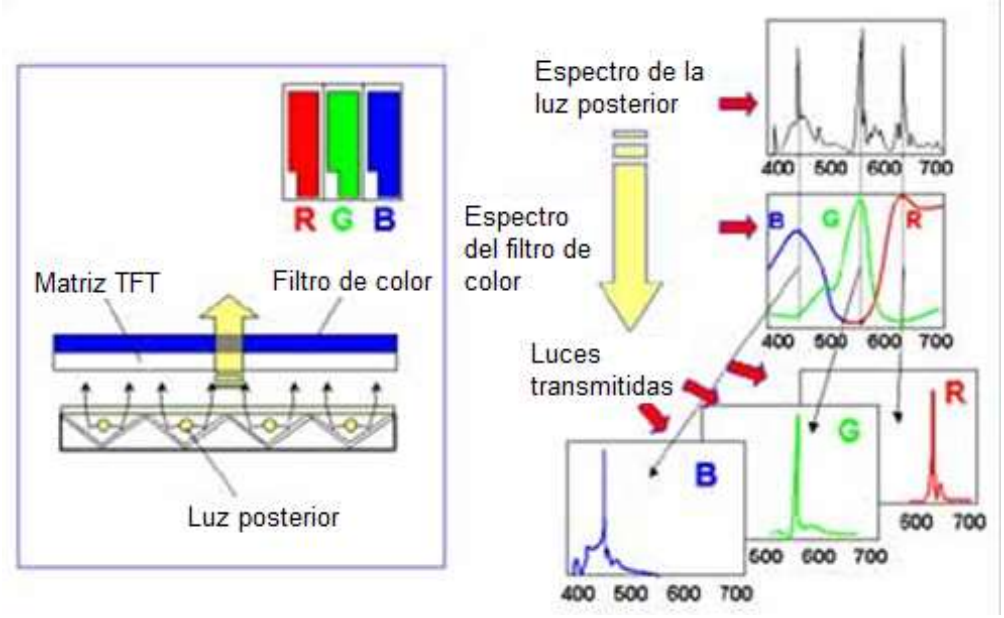


Figura 4.5.8 – Generación de color en un panel LCD TFT

Cada píxel es subdividido en tres subpixeles, pero a la vista se aprecia como uno solo que corresponde a la combinación de los tres colores. Prácticamente cualquier color se puede producir mediante la mezcla de los tres colores primarios.

La cantidad total de colores usando n-bits es 2^{3n}, debido a que cada subpixel puede generar 2^n niveles diferentes de transmitancia. *Info: Samsung Electronics*

4.6 TARJETAS GRÁFICAS

4.6.1 Fundamentos

La tarjeta gráfica es la encargada de generar las señales que controlan el monitor.

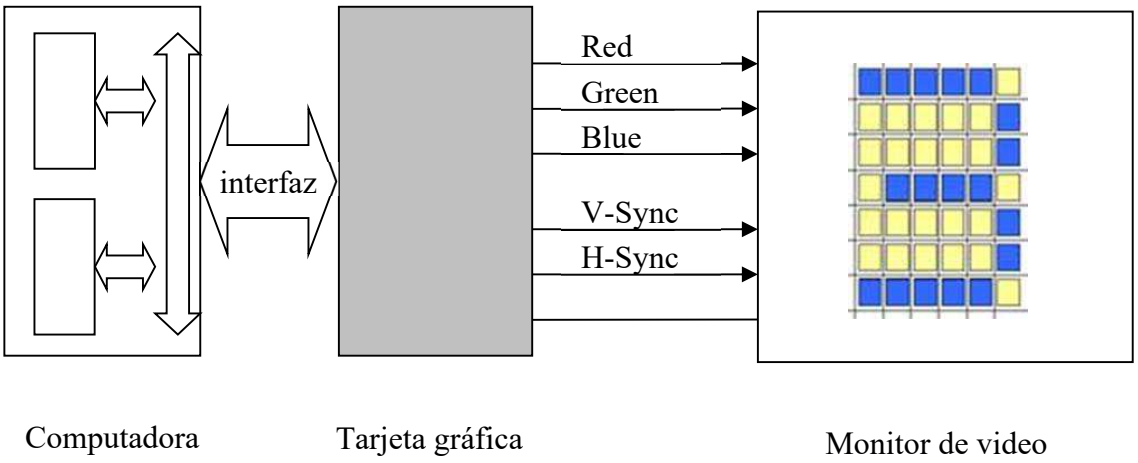


Figura 4.6.1 – Diagrama de conexión de una tarjeta gráfica con interfaz analógica

4.6.2 Evolución

Nombre	Fecha	Resolución	Modo color	Memoria
MDA	1980	80 x 25 (texto) – 640 x 200 (gráfico)	Monocromo	4 KB
CGA	1981	80 x 25 (texto) – 320 x 200 (gráfico)	4 Colores	16 KB
HGC	1982	80 x 25 (texto) – 720 x 348 (gráfico)	Monocromo	64 KB
EGA	1985	80 x 25 (texto) – 640 x 350 (gráfico)	16 Colores	256 KB
VGA	1987	80 x 25 (texto) – 640 x 480 (gráfico)	256 Colores	512 KB
SVGA	1989	80 x 25 (texto) – 1024 x 768 (gráfico)	256 Colores	1 MB
XGA	1992	80 x 25 (texto) – 1024 x 768 (gráfico)	65K Colores	2 MB
SVGA	1998	132 x 60 (texto) – 1280 x 1024 (gráfico)	16,7M Colores	4 MB

Tabla 4.6.1 – tarjetas gráficas 2D

Nombre	Fecha	GPU	Características	Memoria
Voodoo	1997		Antialiasing, Bi linear Filter	4 MB
TNT	1999		Antialiasing, Tri linear Filter, Blur, 6 Mtriángulos/s	4 MB
Geforce 440SE	2002	0:2:4:2 (1)	DX7	128 MB
Geforce 6800GT	2004	6:16:16:16 (1)	DX9c	256 MB
Geforce GTX280	2008	240:80:32 (2)	DX10	512 MB
Geforce GTX1650	2020	896:56:32 (3)	DX12, 3 TFLOPS/s, TDP 75 W, bus 128 bits	4 GB
Geforce RTX3090	2021	10496:328:112:328:82 (4)	DX12, 35 TFLOP/s, TDP 350 W, bus 384 bits	24 GB

Tabla 4.6.2 – tarjetas gráficas 3D

Notas:

(1) Vertex shader: Pixel shader:Texture mapping unit:Render Output unit

(2) Unified Shaders (Vertex shader/Geometry shader/Pixel shader):Texture mapping unit:Render Output unit

(3) Shaders:Texture:Raster

(4) Shaders:Texture:Raster:Tensor core:RT core

A partir de la serie 8000, Nvidia soporta la tecnología CUDA (Compute Unified Device Architecture) que hace referencia tanto a un compilador como a un conjunto de herramientas de desarrollo creadas por nVidia que permiten a los programadores usar una variación del lenguaje de programación C para codificar algoritmos en GPUs de nVidia. Esto permite usar los shaders unificados de la GPU para realizar tareas tradicionalmente destinadas a la CPU. El alto paralelismo que permite la utilización de cientos de procesadores aumenta notablemente la potencia de cálculos específicos.

Dada la potencia de cálculo de los procesadores gráficos, éstos se usan también para procesamiento general, inclusive cálculos muy complejos.

En el apéndice F, figura 11, puede verse una lista de las 10 “supercomputadoras” mas potentes de la actualidad (Julio de 2021) en la que se observa que algunas de ellas sólo utilizan GPUs para lograr su impresionante potencia de cálculo.

4.6.3 Interfases PC-TG

Las tarjetas primeras Tarjetas gráficas se conectaban a los buses habituales de las PC (ISA, PCI, etc.). Luego se desarrolló una interfaz específica para video: AGP (véase 4.3.6) para proveer los crecientes requisitos de velocidad de transferencia que requería el video.

El posterior desarrollo de las GPU y las mejoras en los buses estandarizados hicieron posible, en la actualidad, volver a los buses estándar como el PCI-E. Véanse las características de los distintos buses en este capítulo, en 4.3 Buses en la arquitectura PC compatible.

4.6.4 Interfases TG-Monitor

Desde los comienzos de las tarjetas VGA y aún en la actualidad se usó el estándar VGA, un conector tipo D de 15 contactos con las señales que se muestran en la figura 4.6.4. Con la aparición de los monitores digitales se popularizaron las interfases DVI y recientemente HDMI, que además de video digital incluye sonido.

Figura 4.6.2 – Conectores de video de la TG Geforce GTX 950

Figura 4.6.3 – Aspecto de la TG Geforce GTX 950

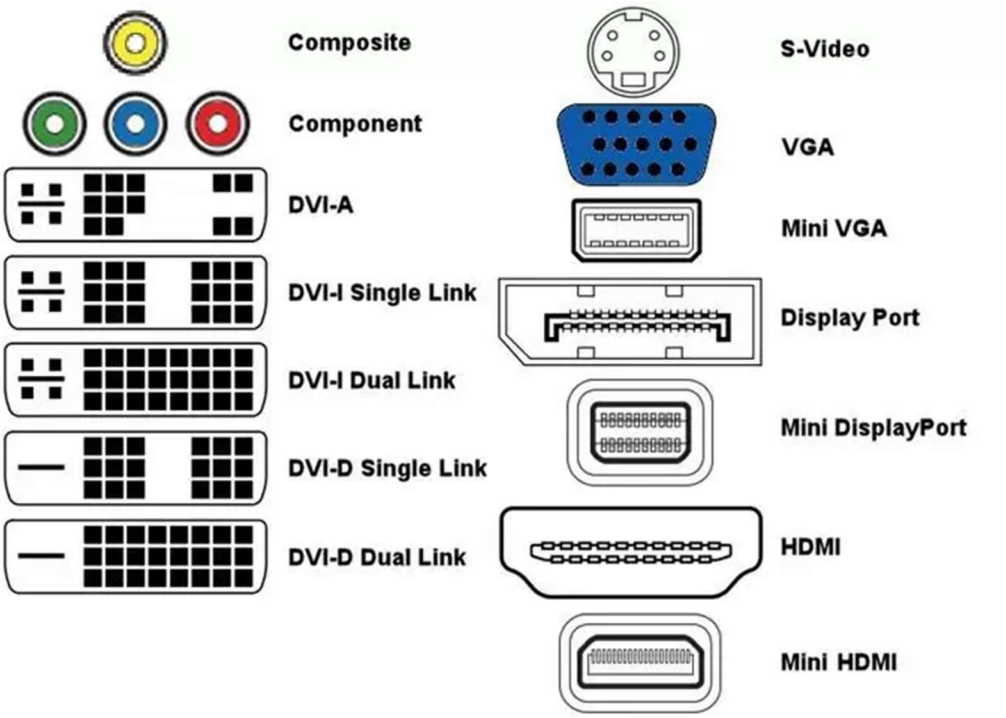


Figura 4.6.4 – Conectores de video

FORMATOS DE VIDEO EN PELÍCULAS							
Formato de Rip	Codec de Video		Codec de Audio	Resolución	Extensión	Tamaño	Calidad
DivX	MPEG-4	DivX	MP3 2.1	640×480	*.avi	700 MB	Baja
DVDrip		XviD	AC3 2.1	720×480	*.avi	700 MB 1.400 MB	Media
BRrip		x264	AAC, AC3 5.1	1280×720 (**720p**) 1920×1080 (**1080p**)	*.mp4 *.mkv	+1.500 MB	Alta

Tabla 4.6.3 Formatos de video

CAPÍTULO 5

DETECCIÓN DE FALLAS Y BENCHMARKS

5.1 DETECCIÓN DE FALLAS

5.1.1 Diagnósticos post

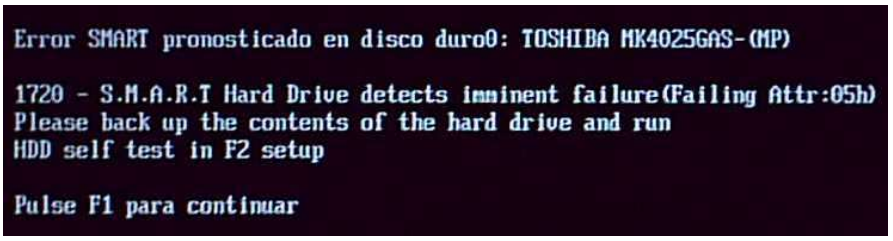
El diagnóstico POST (Power On Self Test) es de los primeros programas que se ejecutan cuando se enciende la computadora. Está cargado en ROM y diseñado específicamente para esa computadora.

El POST verifica que los componentes vitales funcionen correctamente. Si el video funciona, los mensajes de error se muestran en la pantalla. Si no es así, se genera una secuencia diferente de pitidos en el parlante de acuerdo a la falla encontrada. Antes de comenzar cada test, se coloca un código de un byte en un puerto, de manera que se puede saber, con una placa de diagnóstico colocada en el bus, cual es la prueba que dio falla. Si todas las pruebas finalizan con éxito, continúa el arranque normal, cargando el sistema operativo del disco. Desde la configuración de la BIOS se selecciona como ejecutar el POST.

Placa para ver los códigos POST

Quick Power On Self Test	ENABLED reduce el tiempo necesario para realizar el chequeo de arranque (POST). Esto omite ciertos pasos. Es preferible que esté DISABLED para detectar posibles problemas durante el arranque y no mientras se trabaja.
Halt On	Durante el auto chequeo al encender el ordenador (POST), la BIOS se detiene si detecta algún error de hardware. Se puede indicar a la BIOS que ignore ciertos errores y continúe el proceso de arranque. Estas son las posibilidades: No errors — No para en ningún error. All errors — Si se detecta algún error, se detiene el arranque y se pide que se corrija el error. All, But Keyboard — Se detiene en todos los errores excepto el de teclado. All, But Diskette — Se detiene en todos los errores excepto el de diskette All, But Disk/Key — Se detiene en todos los errores excepto el de teclado o disco.

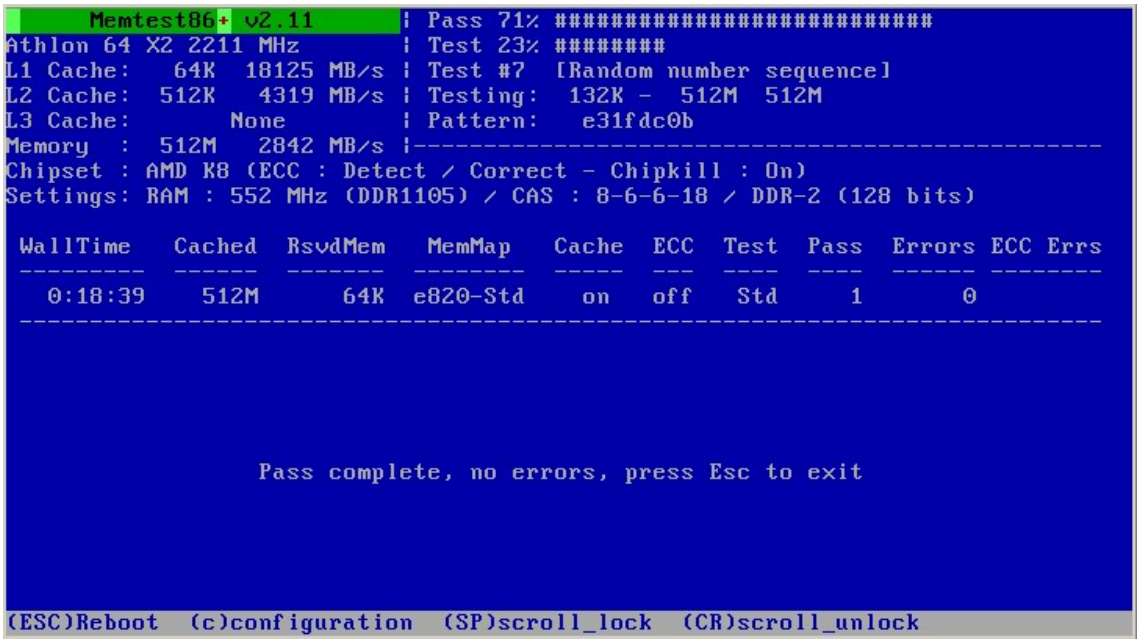
Desde el POST se chequea también la función S.M.A.R.T de los discos rígidos. Si se detecta alguna anormalidad se informa como se ve en la figura siguiente.



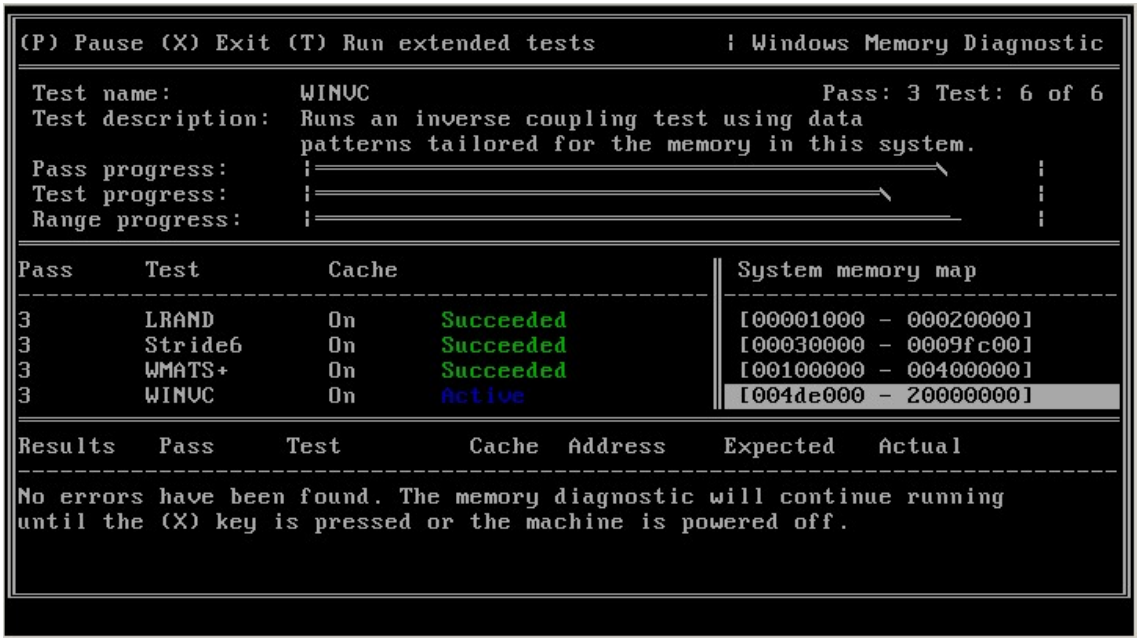
5.1.2 Diagnósticos Stand-Alone – RST, Memtest86 y WMD

Introducción

Se dice que un diagnóstico es “stand-alone” cuando puede ejecutarse cargándose a sí mismo sin necesidad del sistema operativo. Se requiere una mínima parte del hardware en estado funcional. Se pueden mencionar 3 diagnósticos de memoria de este tipo: RST (Ram Stress Test), Memtest86+ y Windows memory diagnostics. Se pueden ejecutar desde un diskette, un cd o un pen drive autoarrancables. Al reiniciar la computadora, la pantalla (en modo texto) muestra “loading…” y luego inmediatamente se inicia la ejecución con los valores por defecto. En este ejemplo la máquina tiene 512 MB de RAM.



Pantalla de Memtest86+



Pantalla de Windows memory diagnostics

5.1.3 Diagnósticos para DOS – PC-Check

Introducción

Aunque un diagnóstico que se ejecute bajo DOS presupone que una parte importante del hardware funciona, este tipo de programas es muy apropiado para garantizar que este hardware es estable en condiciones de exigencia extrema para un componente. Con un software de este tipo se puede establecer una metodología de testeo de equipos recién armados, recién reparados o aquellos que presentan fallas que detecta el usuario con las aplicaciones propias de su trabajo.

El programa que se analizará a continuación, **Pc-Check**® de Eurosoft, está compuesto de un módulo de **información**, uno de **diagnósticos,** uno de **burn-in (1)** y varias utilidades mas. El CD es auto arrancable por lo que no necesita la carga previa de un sistema operativo. En la configuración de arranque de la PC se debe seleccionar el medio donde está almacenado el programa Pc-Check, en nuestro caso la unidad de CD. También existen versiones que arrancan desde un pen-drive USB.

Cuando el programa se inicia, muestra la siguiente pantalla:

Menú principal

Pc-Check® Este programa es comercial y no puede ser distribuido ni utilizado si no se posee la correspondiente licencia del fabricante. Las copias de pantalla que se muestran en este apunte son sólo con fines didácticos. La página del fabricante es: http://www.eurosoft-uk.com/pccheck.html.

(1) Burn-in es el proceso mediante el cual un componente o un sistema es ejercitado antes de que sea puesto en servicio. El objetivo es detectar aquellos componentes que podrían fallar a causa de la alta tasa de fallas de la parte inicial de la "curva de la bañera" característica de la confiabilidad de los sistemas. Si el período de "burn-in" es suficientemente extenso, se puede confiar que el sistema o componente no fallará cuando sea puesto en servicio normal.

```
Pc-Check 6.21 (c) Eurosoft (UK) Ltd 1988-2008

                System Information Menu

 - Information ---------------   - Utility ----------------
 System Overview                 I/O Port Browser
 Hardware ID Image               Memory Browser
 System Management Info          Sector Browser
 PCI Bus Info                    CPU Freq. Monitor
 IDE Bus Info                    CMOS RAM Utils
 PCMCIA/CardBus Info             SCSI Utils
 Interrupt Vectors               Text File Editor
 IRQ Routing Info
 Device Drivers                  - Settings ----------------
 APM Info                        Start-Up Options
                                 Change Activation

Use ←↑↓→ to Move Bar, <ENTER> to Select, <ESC> to Exit
```

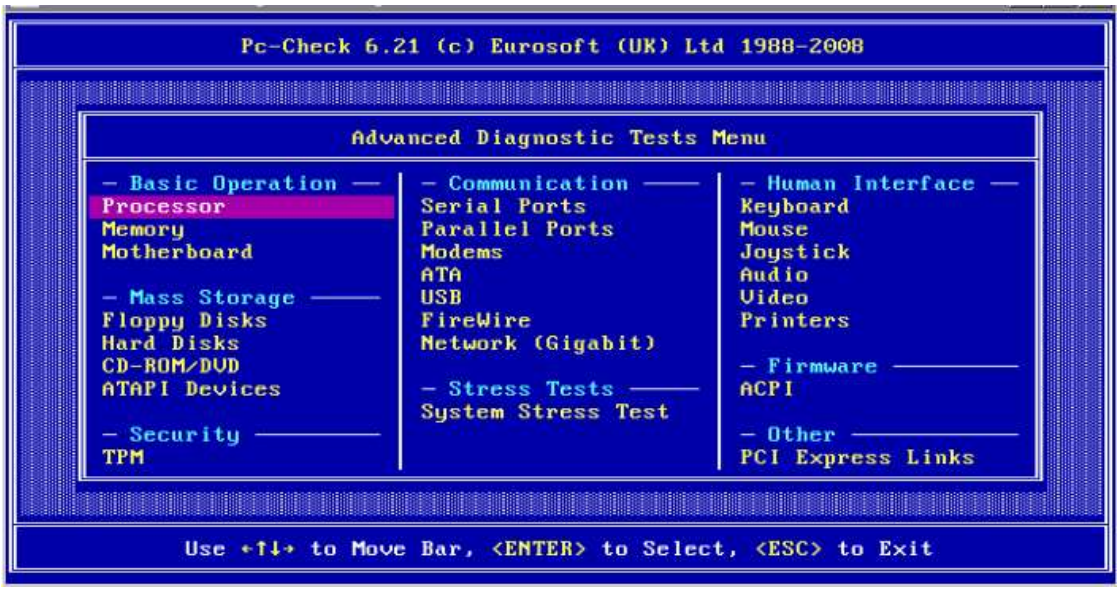
Menú de información del sistema

```
                Pc-Check - System Overview

System ......................... Manufacturer: innotek GmbH²
                                 Product: VirtualBox²
                                 Version: 1.2²
                                 Serial: 0²

BIOS ........................... Vendor: innotek GmbH²
                                 Release: 06/23/99 (12/01/2006²)

Processor ...................... Type: AMD Athlon(tm) 64 X2 Dual Core Proces...
                                 Speed: 2210.43 MHz
                                 Multi-Core: Present, 2 way
                                 Hyper-Threading: Not Present
                                 Total Logical Processors: 2

Memory ......................... Available: 1023MB
                                 Device Type: Not specified²

Cache (per core) ............... Level 1: Disabled
                                 Level 2: 512KB                          ▼

     Use ↑↓ <PgUp> <PgDn> <Home> <End> to Scroll, <ENTER> to Continue
```

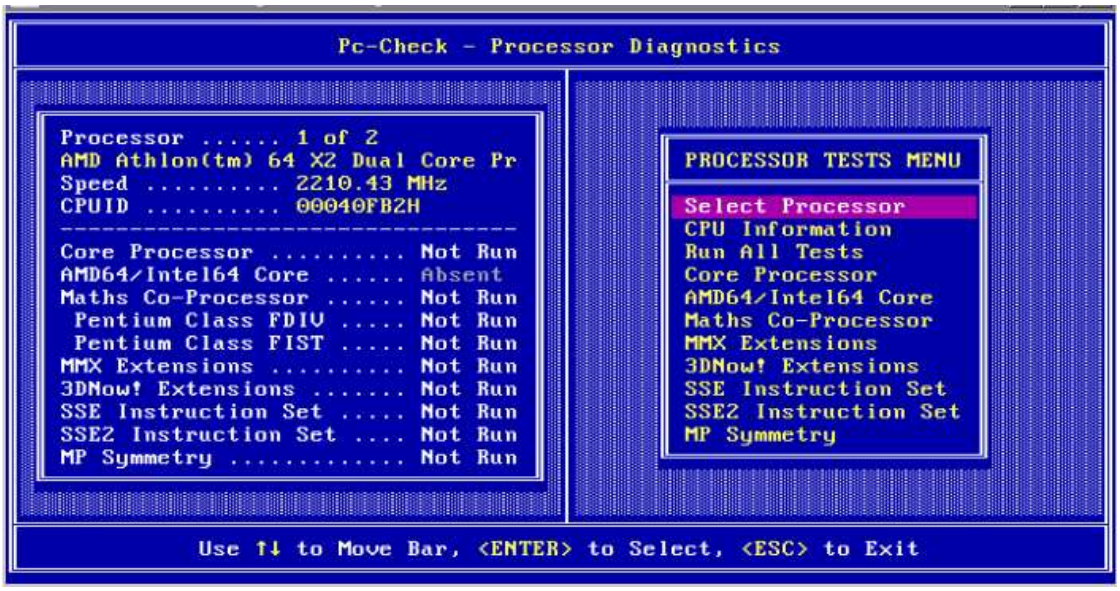
Vista general del sistema (página 1)

```
                Pc-Check - System Overview

Cache (per core) ............... Level 1: Disabled                      ▲
                                 Level 2: 512KB

Drives (fixed media) ........... VBOX HARDDISK (1.07GB)

Drives (removable media) ....... VBOX CD-ROM (ATAPI, CD/DVD),
                                 VBOX CD-ROM (ATAPI, CD/DVD)

Video .......................... VirtualBox VBE BIOS http://www.virtualbox.o...

Audio .......................... 82801AA 8xx Chipset AC'97 Audio Controller

Network ........................ 82540EM Gigabit Ethernet Controller

Buses .......................... PCI, ISA/LPC, USB, IDE

Ports .......................... Serial

Information marked with a ² is provided by the BIOS and may be inaccurate.

     Use ↑↓ <PgUp> <PgDn> <Home> <End> to Scroll, <ENTER> to Continue
```

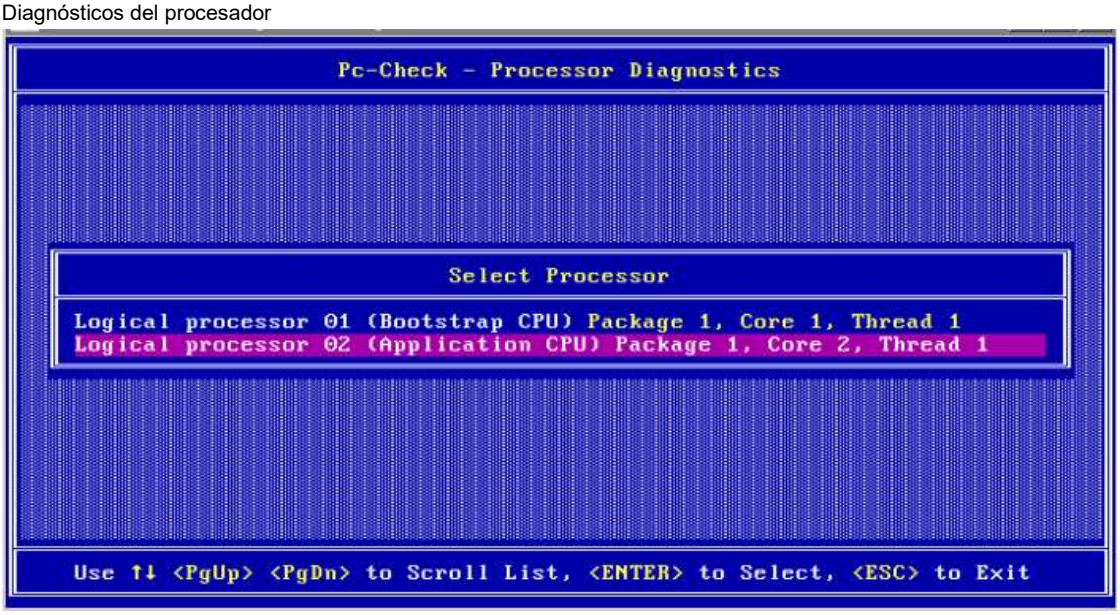
Vista general del sistema (página 2)



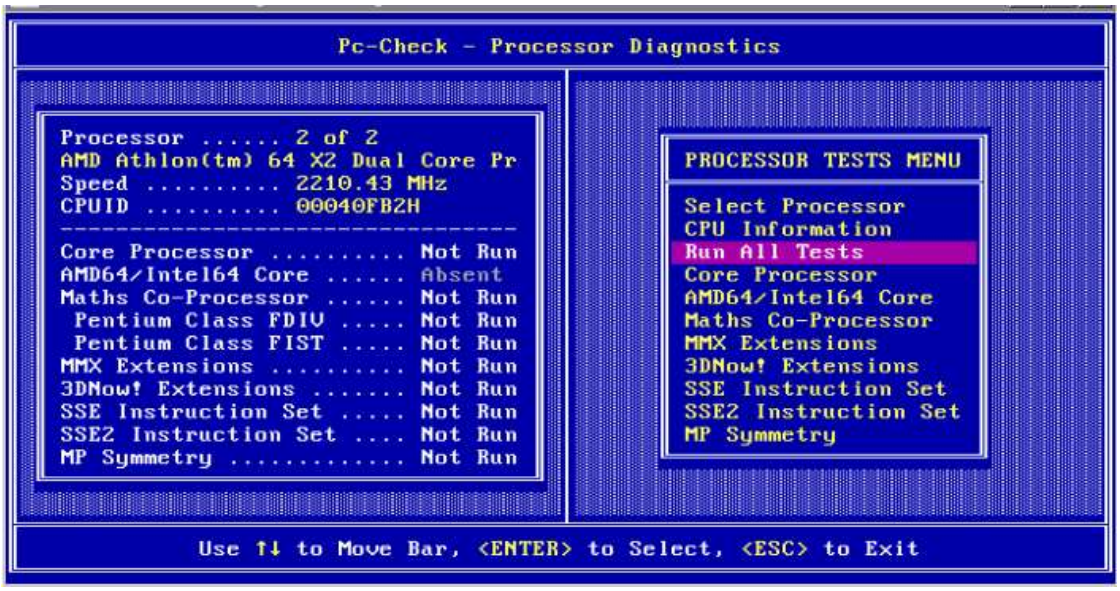
Menú de diagnósticos avanzados



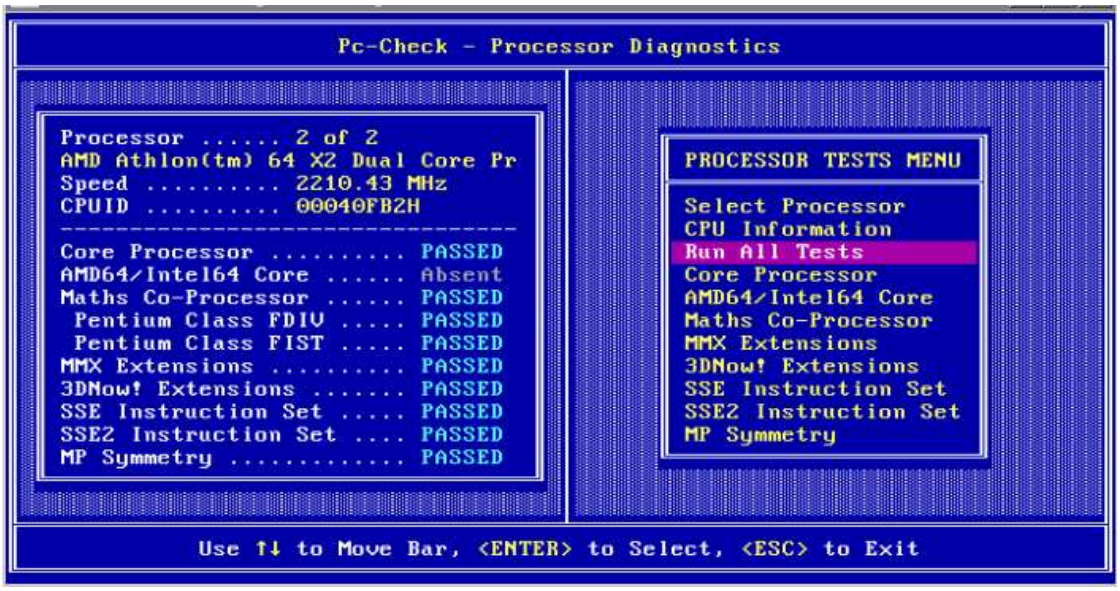
Diagnósticos del procesador



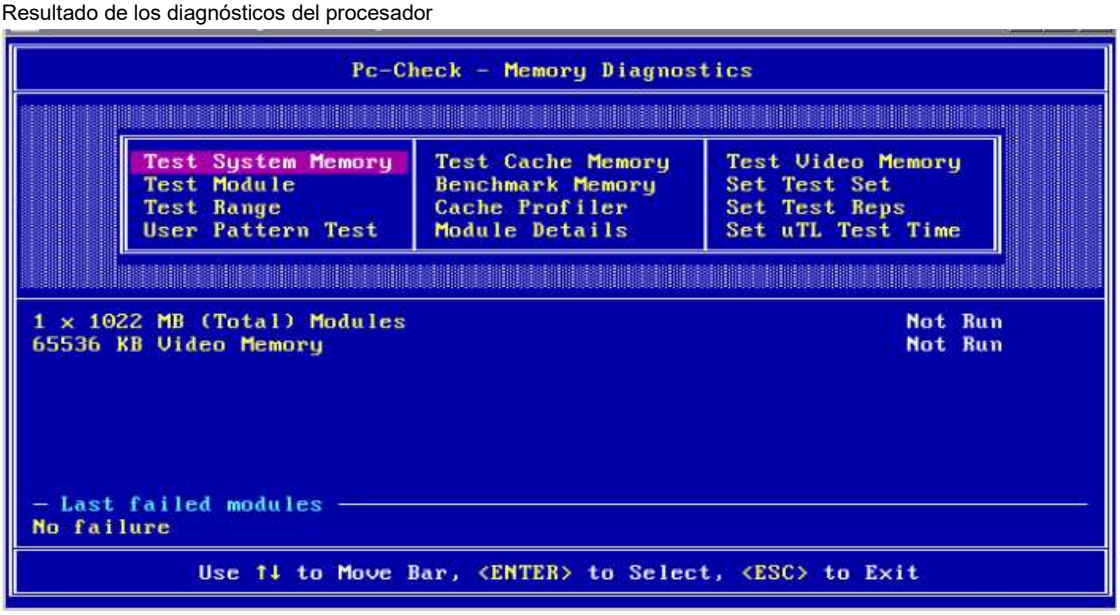
Selección del procesador (núcleo)



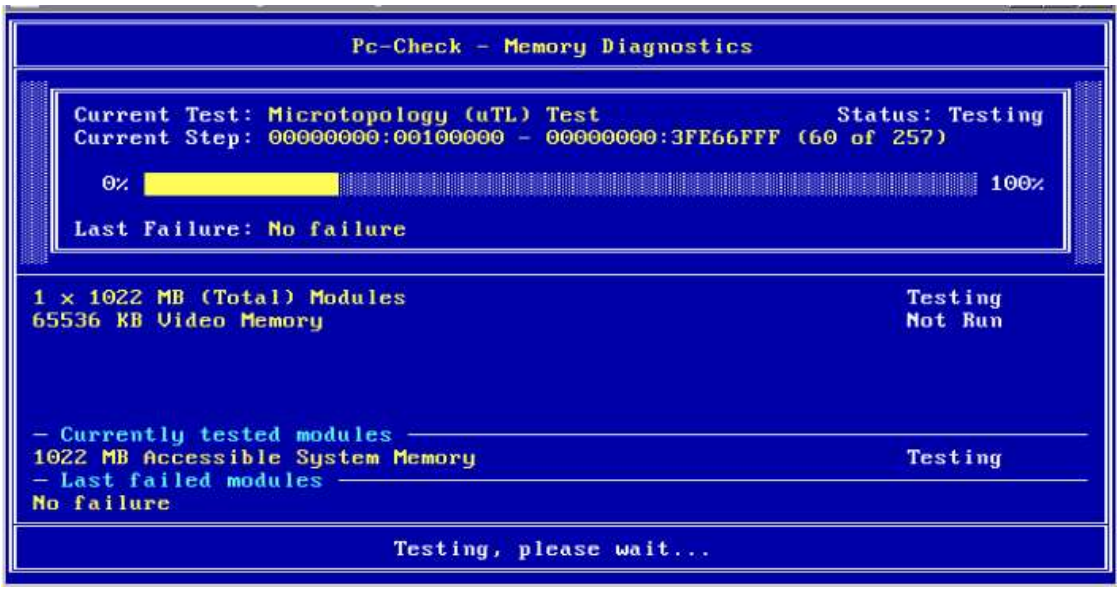
Selección del diagnóstico a ejecutar (todos)



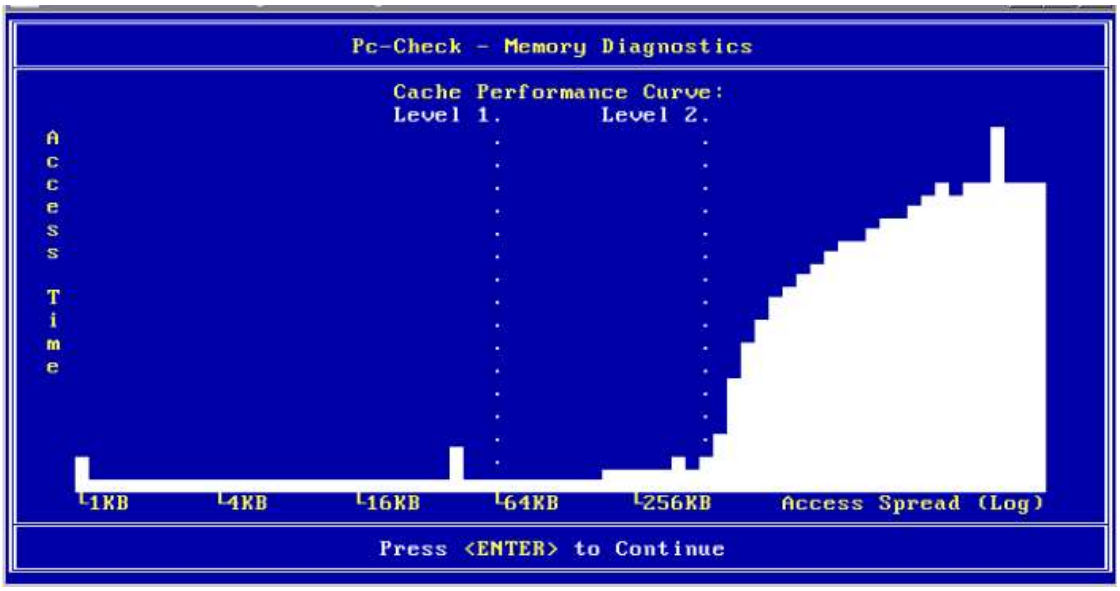
Resultado de los diagnósticos del procesador



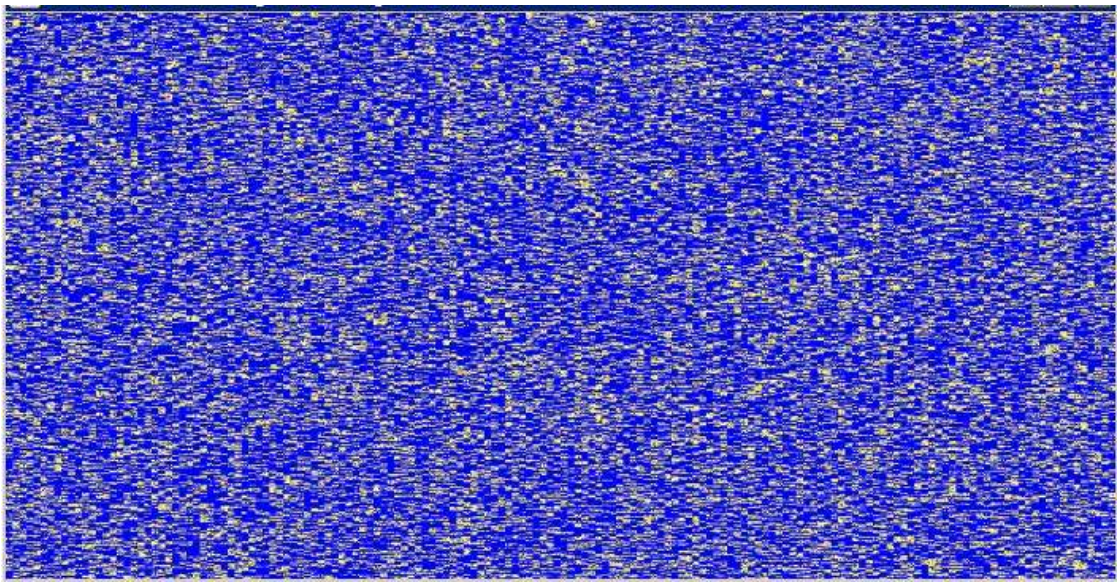
Diagnósticos de memoria (selección)



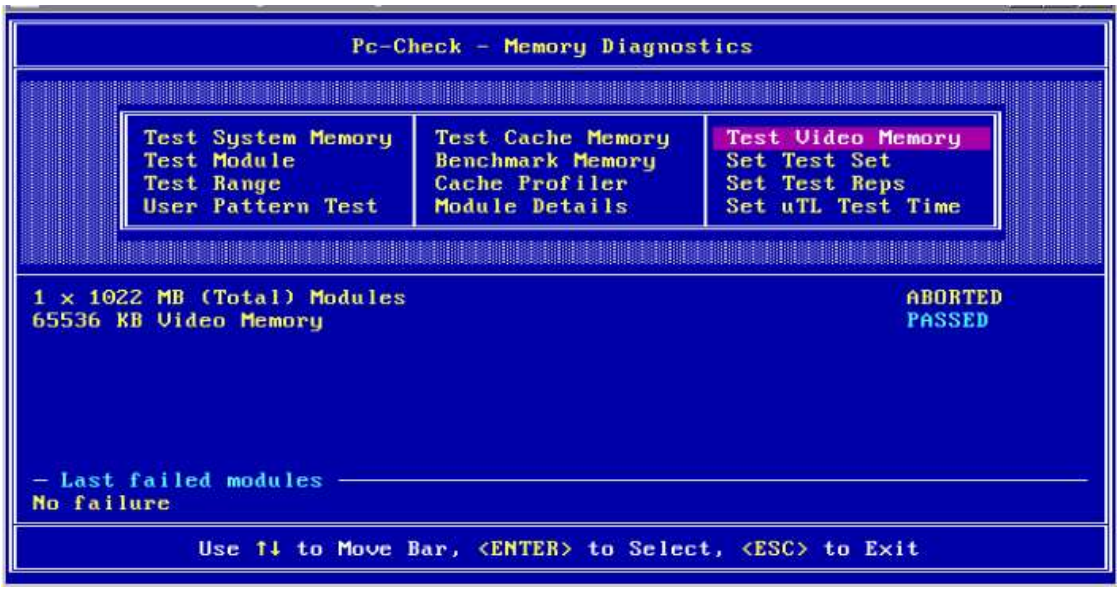
Diagnósticos de memoria del sistema (en ejecución)



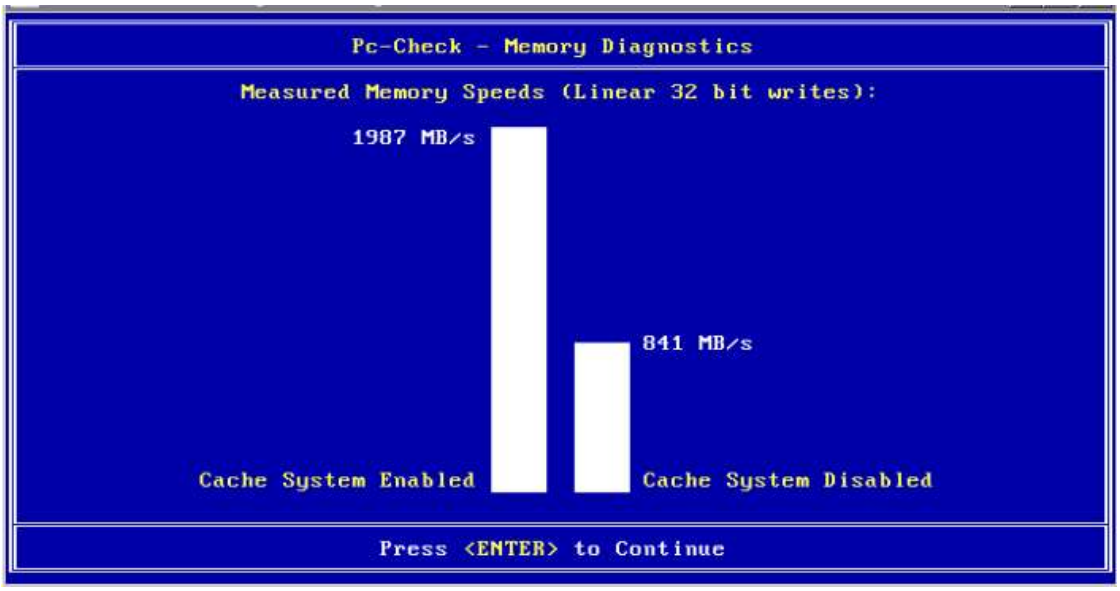
Performance de la memoria cache



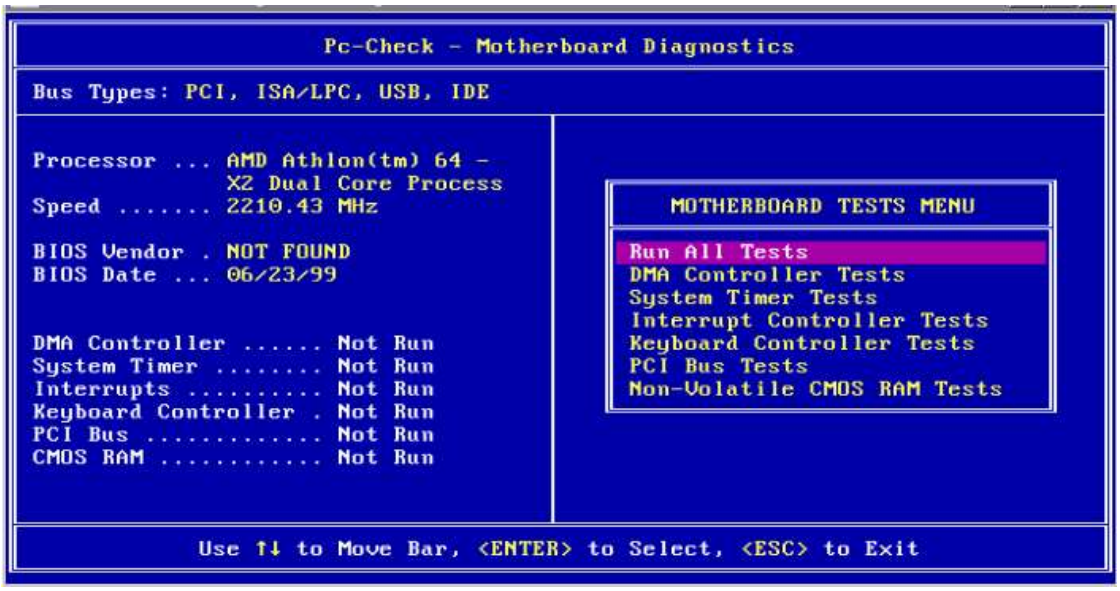
Diagnósticos de memoria de video (en ejecución)



Diagnósticos de memoria de video (finalizado)

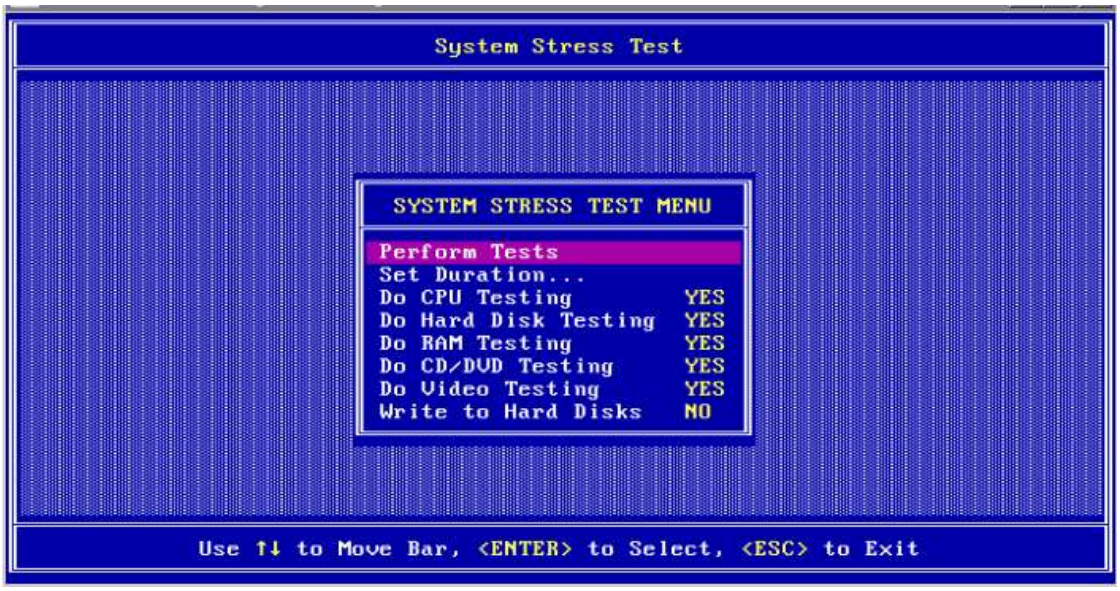


Performance de la memoria cache habilitada vs deshabilitada

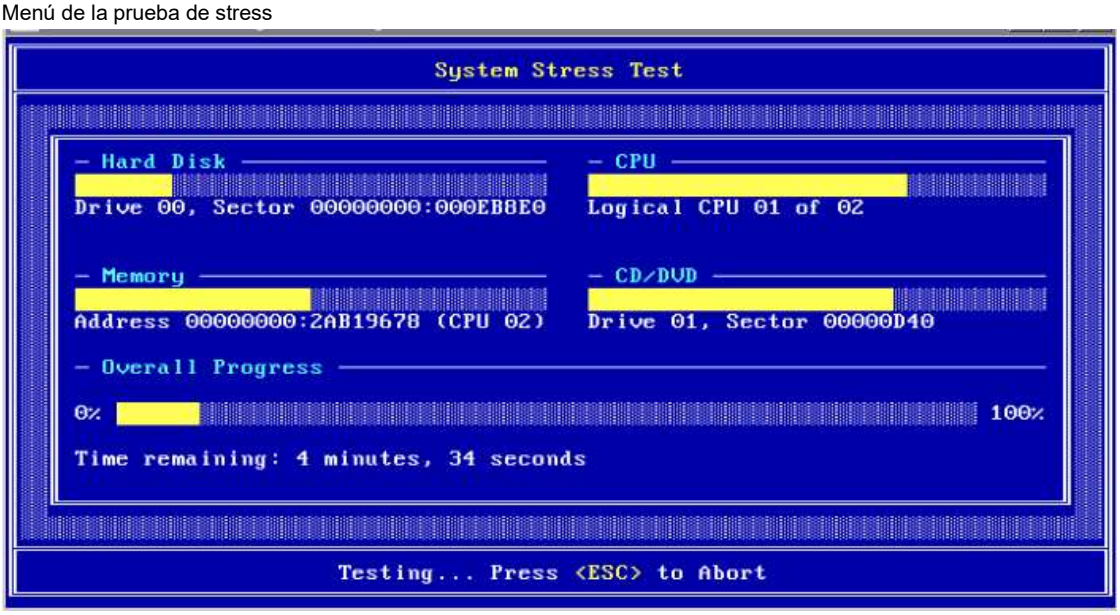


Diagnósticos de la tarjeta principal (selección)

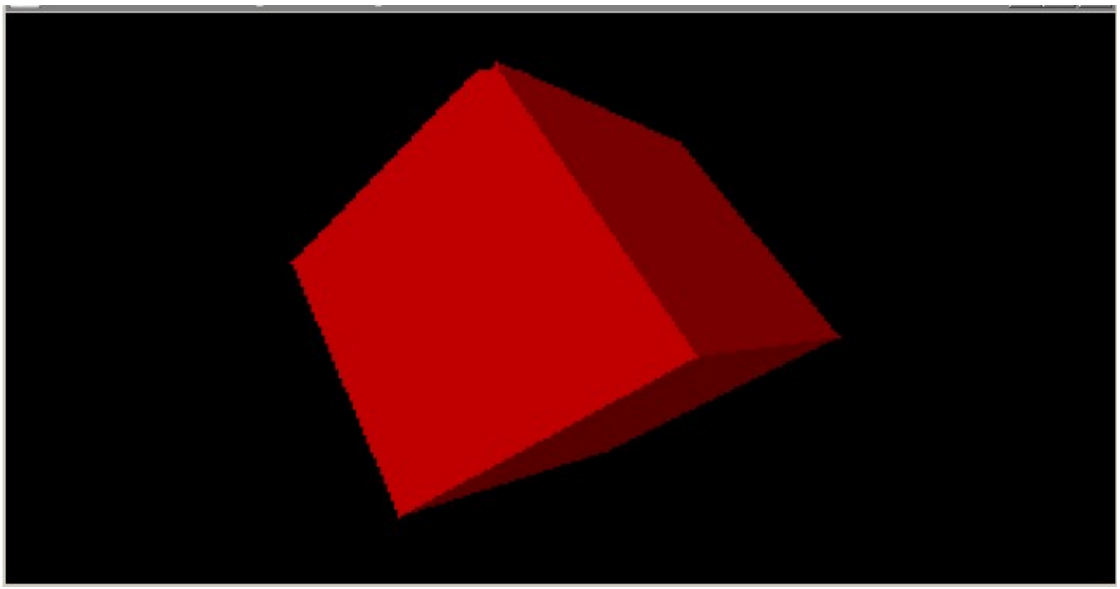
Diagnósticos de la tarjeta principal (finalizado)



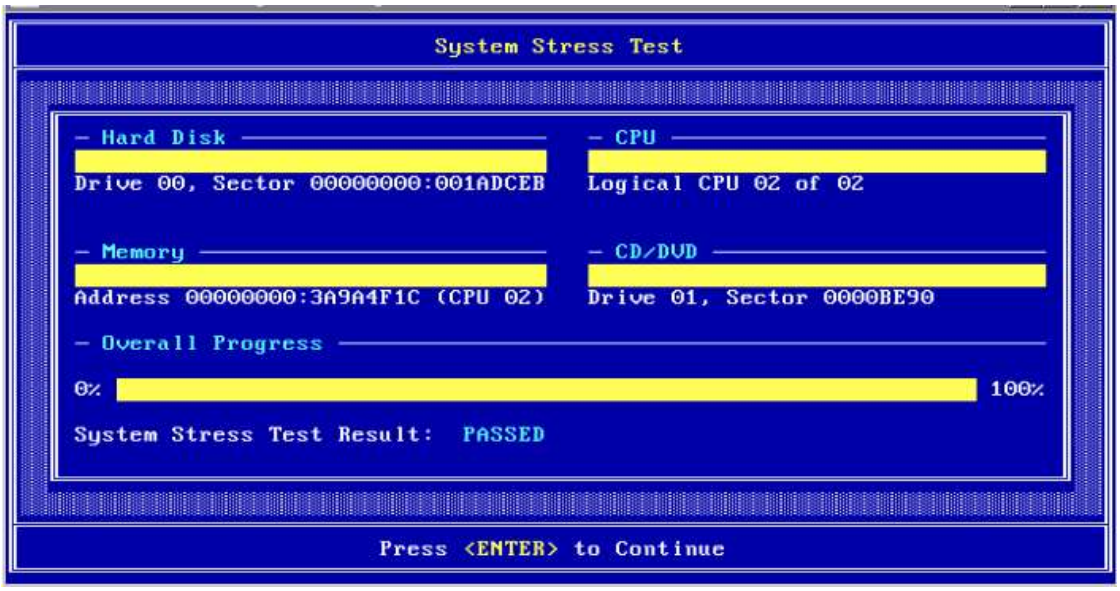
Menú de la prueba de stress



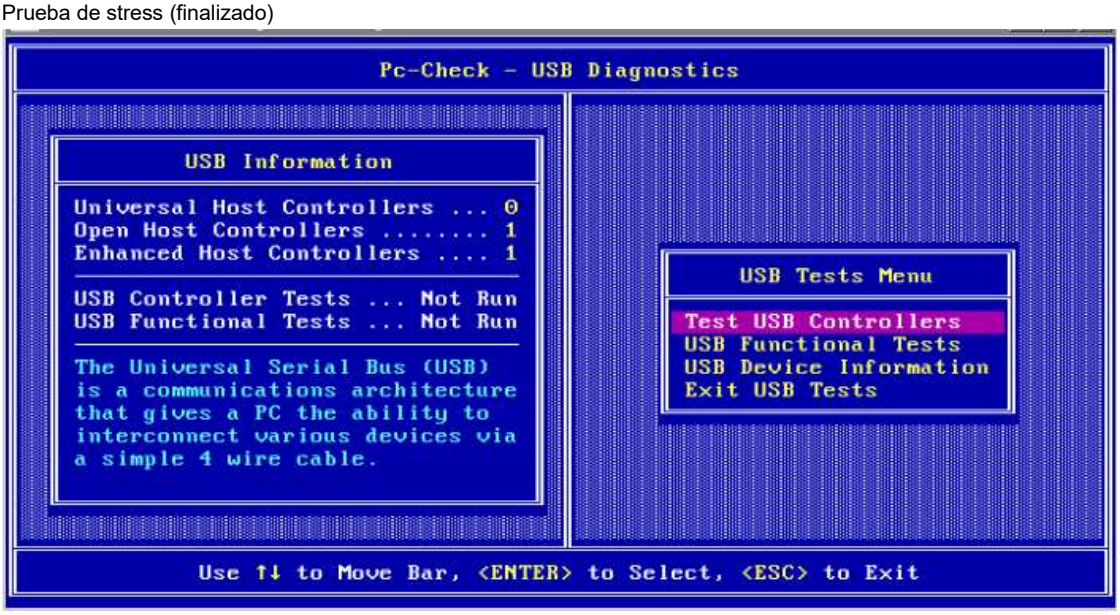
Prueba de stress (en ejecución: disco rígido, CPU, Memoria, DVD)



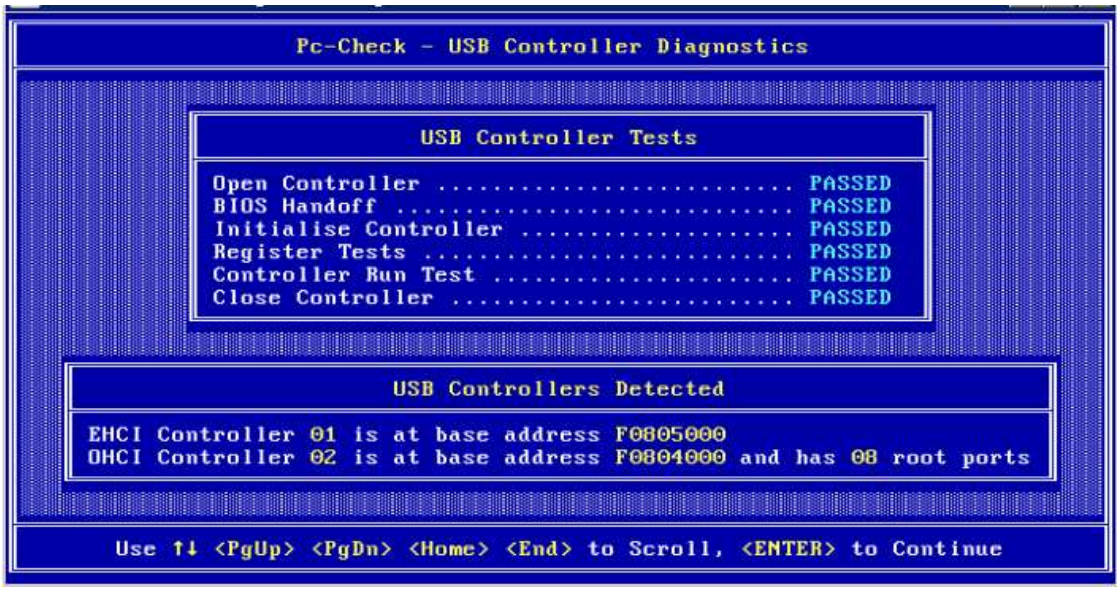
Prueba de stress (en ejecución: video)



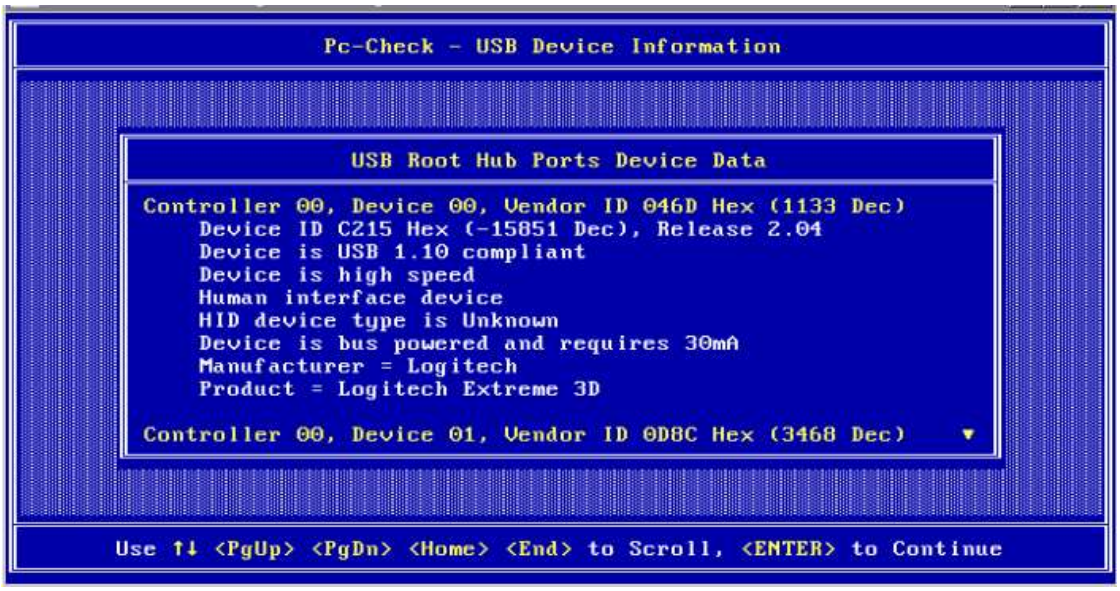
Prueba de stress (finalizado)



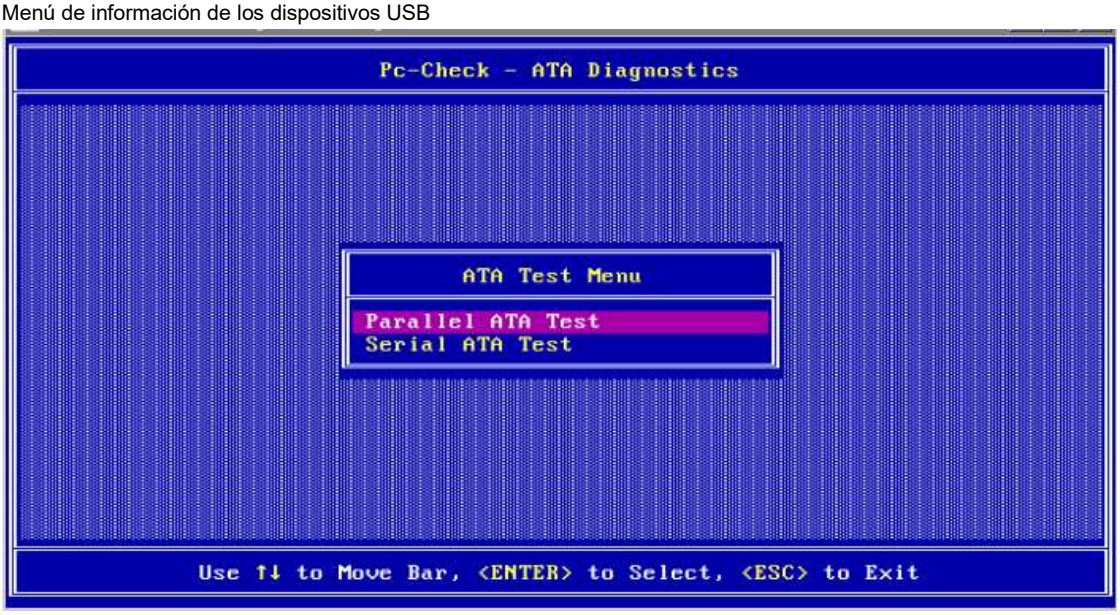
Menú de los diagnósticos USB (selección)



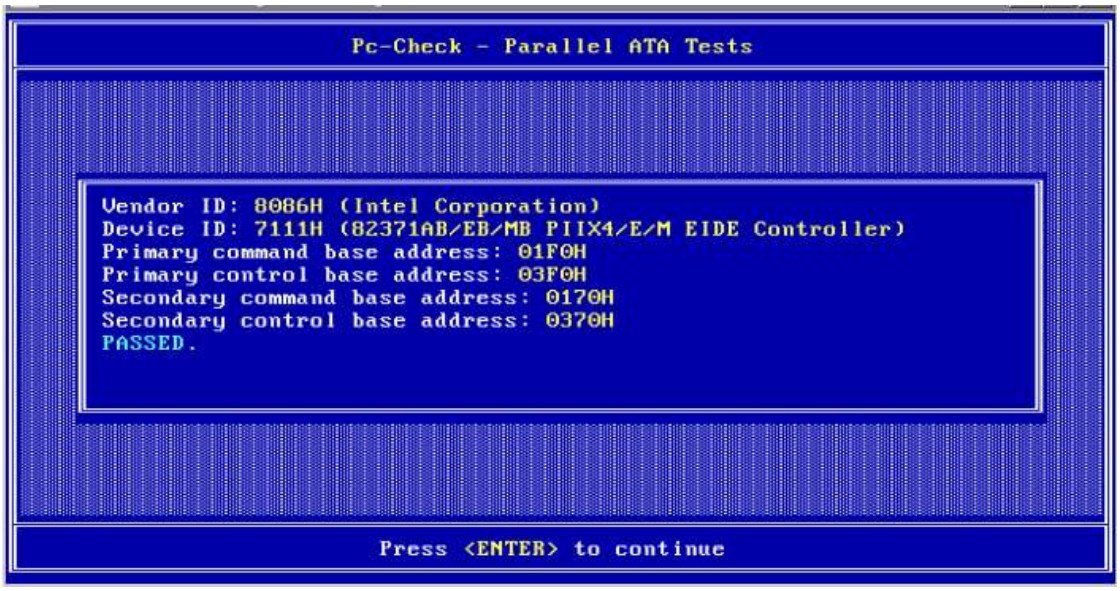
Menú de los diagnósticos USB (finalizado)



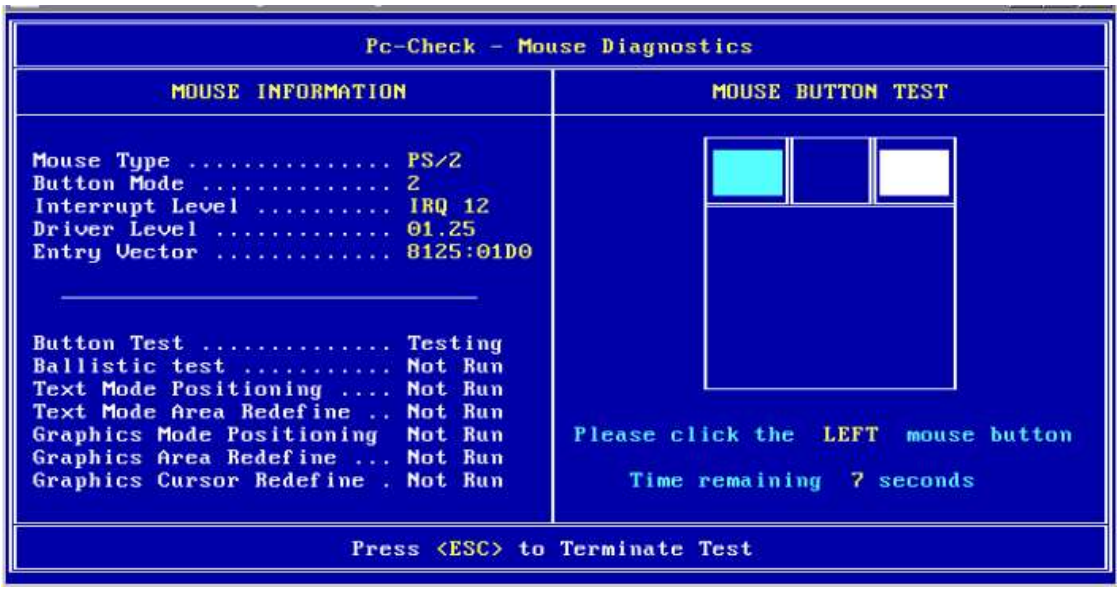
Menú de información de los dispositivos USB



Menú de los diagnósticos de discos (PATA y SATA)

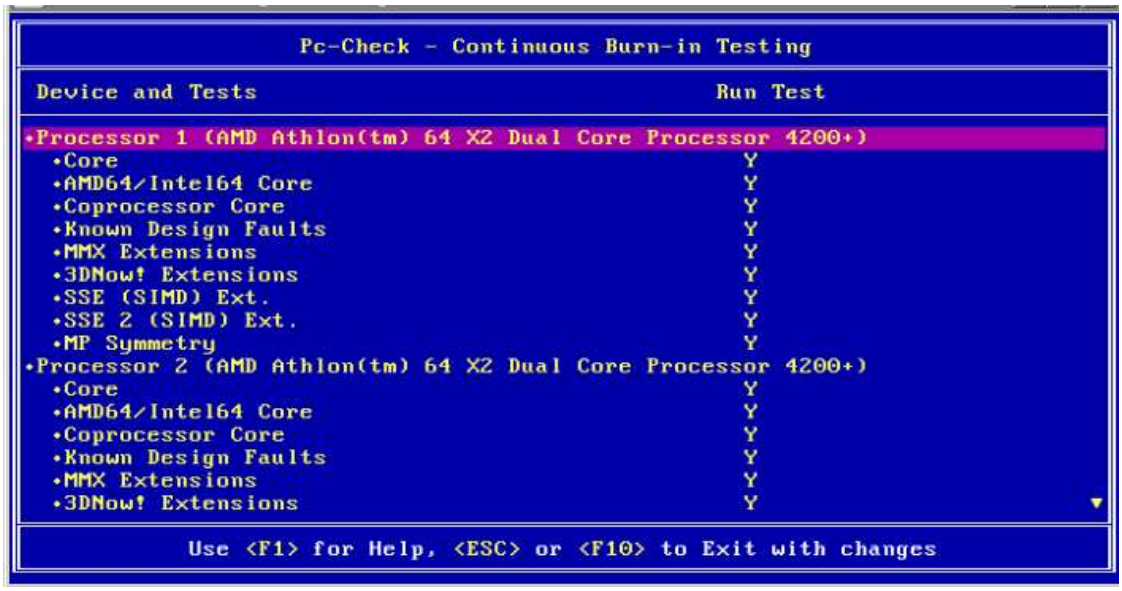


Diagnósticos de discos PATA (finalizado)



Diagnósticos de mouse (en ejecución)

Menú de diagnósticos Burn-in



Diagnósticos Burn-in (selección de los dispositivos a ejercitar)

Para realizar un diagnóstico profesional se debe contar con los “loop-backs” para conectar en el puerto serie, paralelo, etc. Estos tienen un conector y un cableado que permiten que las señales de salida se reenvíen como entrada, permitiendo así verificar el funcionamiento hasta al mismo conector.

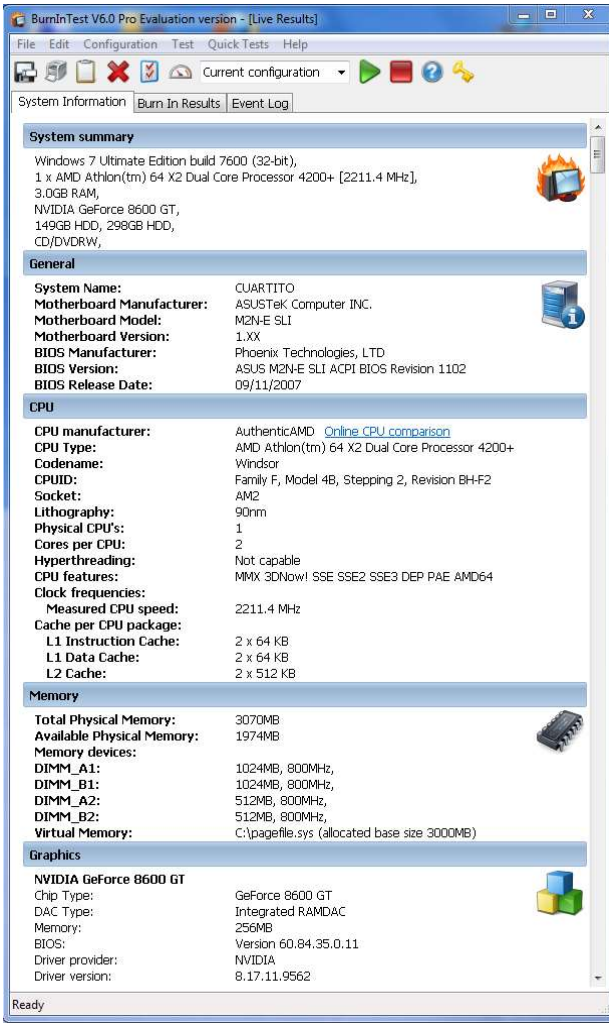
Loop-back USB

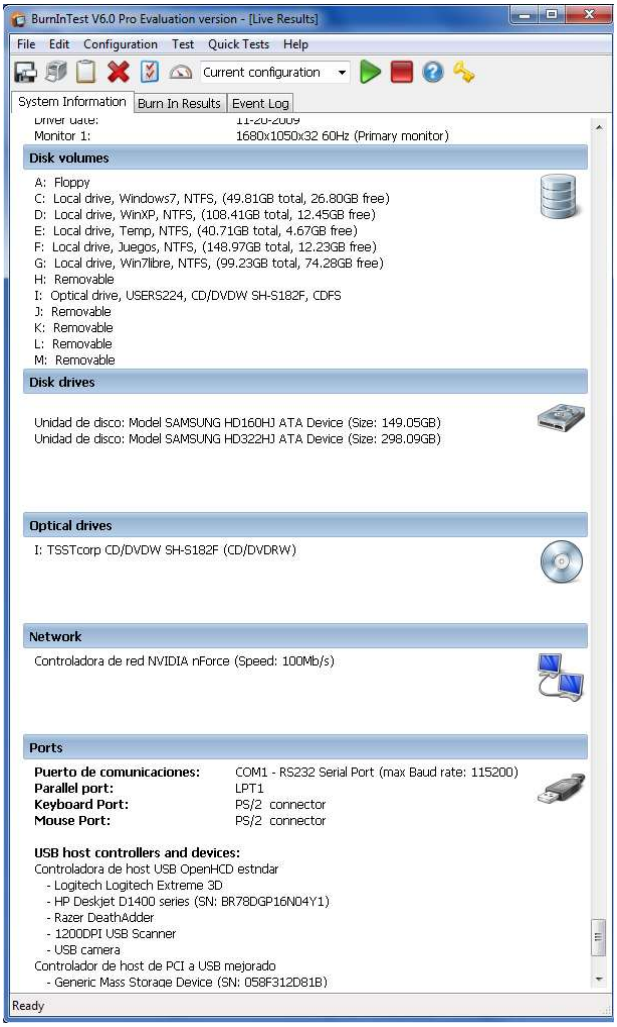
Loop-backs serie y paralelo

5.1.4 Diagnósticos para Windows – Passmark BurnIn Test

Aunque un diagnóstico que se ejecute bajo Windows presupone que una parte importante del hardware funciona, este tipo de programas es muy apropiado para garantizar que este hardware es estable en condiciones de exigencia extrema para un componente, en la situación en la que todos los componentes funcionan simultáneamente, para probar todos los puertos, etc. Con un software de este tipo se puede establecer una metodología de testeo de equipos recién armados, recién reparados o aquellos que presentan fallas que detecta el usuario con las aplicaciones propias de su trabajo.

El programa que se analizará a continuación, BurnIn Test de Passmark está compuesto de un módulo de **información** y uno de **diagnósticos**. (http://www.passmark.com). En el módulo de información se puede ver un resumen de los componentes del equipo.





Como se puede apreciar, también identifica los periféricos conectados a los puertos USB, ya que esta interfaz permite consultar sus nombres a los propios dispositivos. El diagnóstico, de resultar exitoso, confirmará la funcionalidad y estabilidad tanto del **hardware** como del **sistema operativo**, en este caso Windows 7.

Una vez seleccionados los componentes que deseamos probar, se inicia el módulo de diagnósticos, que muestra una serie de ventanas correspondientes a cada dispositivo.

La primera corresponde a la prueba de la CPU, donde se ve el avance medido en millones de operaciones. Debajo de ésta se ve la prueba de la memoria RAM, que consiste en grabar y leer distintos patrones de bits en la mayor parte de la memoria. En este caso, de los 3 GB que tiene el sistema, se probarán 2 GB. Recordemos que para una prueba intensiva de la memoria es mas confiable el diagnóstico RST, que prueba la totalidad y con patrones mas exigentes, arrancando sin necesidad de otro sistema operativo.

Debajo se ve la prueba de la memoria de video (2D) y por último la prueba de la GPU de video (3D). Los relojes son "gadgets", que superpuse para ver la ocupación de la CPU, el uso de memoria y el reloj de Windows mientras se ejecuta el test.

En la columna del centro, se ve el avance del test de sonido, que por supuesto deberá confirmarse con la audición por un auricular, los parlantes del sistema conectados a las salidas de audio de la computadora o un loop-back y un software específico.

A continuación, está el avance de la prueba de la placa de red.

Si se seleccionó un video opcional, se lo podrá ver en la ventana siguiente.

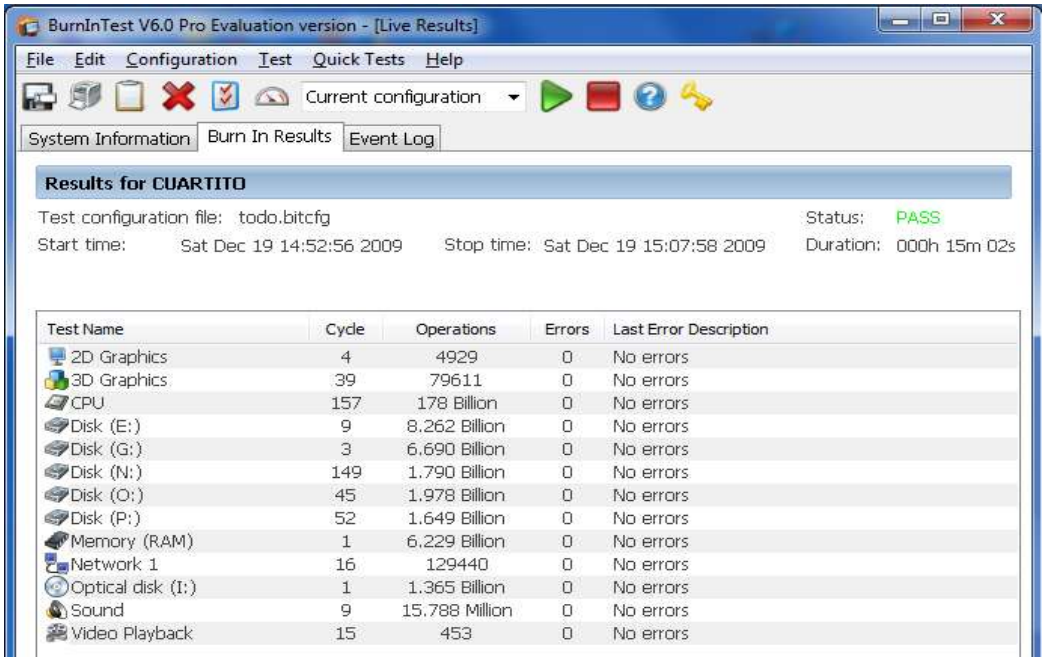
Al final de esta columna se ve el avance de una prueba de lectura de un DVD de datos que se colocó previamente en la respectiva unidad de accionamiento. Desde el menú de opciones también se puede elegir la grabación de un CD o DVD apropiado.

En la columna de la derecha se ve el avance de los diagnósticos de medios de almacenamiento masivo que se hayan seleccionado. En este caso el primero corresponde a una partición de uno de los discos rígidos, la ventana de abajo es una partición del otro disco rígido, y las tres restantes son 3 Pen-drives conectados en los 3 puertos USB frontales del gabinete.

La versión de BurnIn ejecutada aquí es de prueba y limitada a 15 minutos por corrida. La versión comercial permite probar los equipos el tiempo que el ingeniero crea necesario (usualmente 2 a 6 horas) dependiendo de la confiabilidad pretendida si es un equipo nuevo, una reparación, etc. Podría ser mas tiempo si se diagnosticó una falla aleatoria.

Si se trata de una búsqueda de fallas direccionada a un componente en particular, se puede seleccionar sólo éste, de manera que la prueba sea mas exigente con el y no "desgaste" innecesariamente el resto del equipo.

Por último, el programa genera un reporte, que se puede ver en la figura siguiente:



Test Name	Cycle	Operations	Errors	Last Error Description
2D Graphics	4	4929	0	No errors
3D Graphics	39	79611	0	No errors
CPU	157	178 Billion	0	No errors
Disk (E:)	9	8.262 Billion	0	No errors
Disk (G:)	3	6.690 Billion	0	No errors
Disk (N:)	149	1.790 Billion	0	No errors
Disk (O:)	45	1.978 Billion	0	No errors
Disk (P:)	52	1.649 Billion	0	No errors
Memory (RAM)	1	6.229 Billion	0	No errors
Network 1	16	129440	0	No errors
Optical disk (I:)	1	1.365 Billion	0	No errors
Sound	9	15.788 Million	0	No errors
Video Playback	15	453	0	No errors

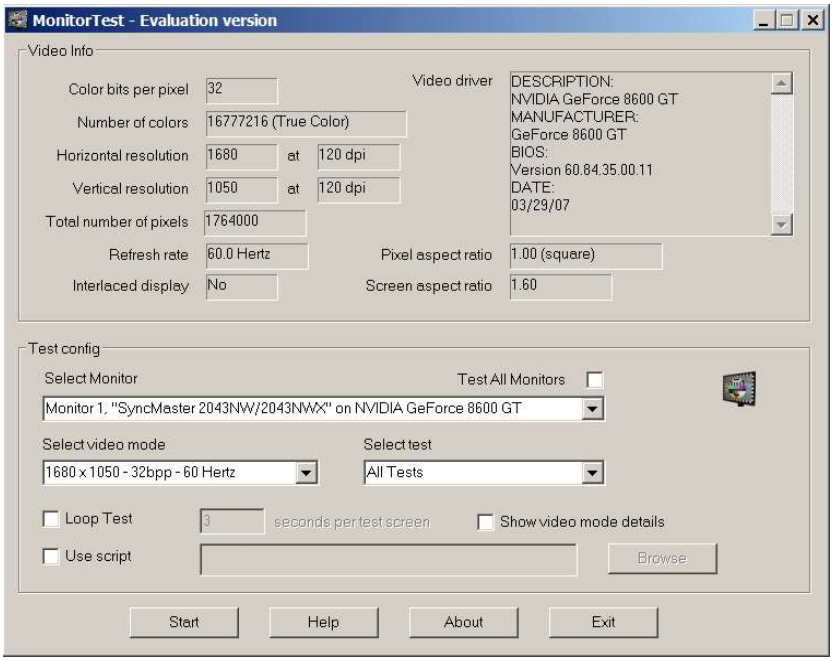
También se crea un reporte en formato texto donde detalla la instalación, los tests realizados, los errores encontrados y toda otra información que permite documentar el trabajo. Se puede personalizar el reporte en formato texto para que tenga la apariencia de un certificado, con logos, etc. lo que le da un aspecto profesional y hace a la imagen de confianza que debe tener un fabricante o un proveedor de servicios de reparaciones.

Este programa se puede instalar en la PC o se puede ejecutar desde un pen-drive y existen versiones para SO de 32 y 64 bits, tanto para Windows como para Linux.

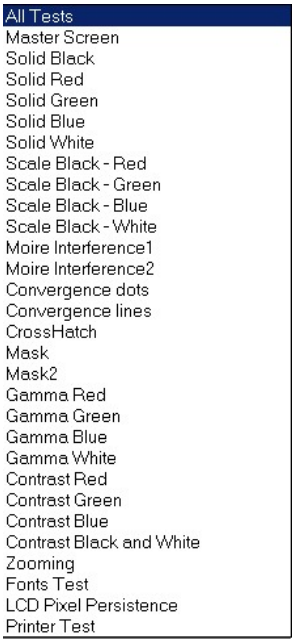
Se pueden adicionar módulos para periféricos no incluidos en la provisión original. En este caso, puede ser mas conveniente obtener programas especiales de probada eficacia, como por ejemplo aquellos para calibrar pantallas de video, para uso en diseño gráfico.

5.1.5 Diagnósticos para Windows – Passmark Monitor Test

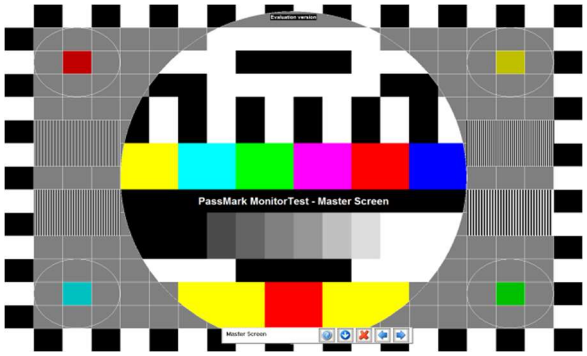
Existen varios programas para ayudarle a mejorar y calibrar sus ajustes, con material de test como diagramas de color y patrones de prueba. Uno de ellos es **"Passmark Monitor Test"**, cuya versión de prueba se verá a continuación.



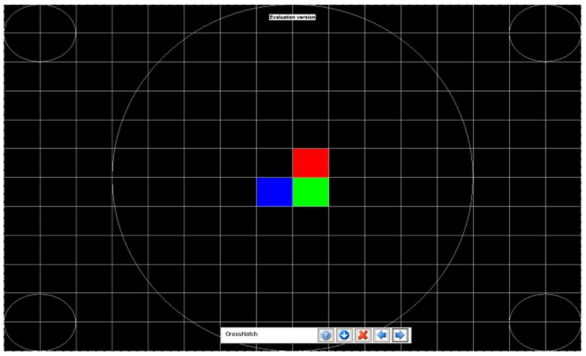
Pantalla de inicio



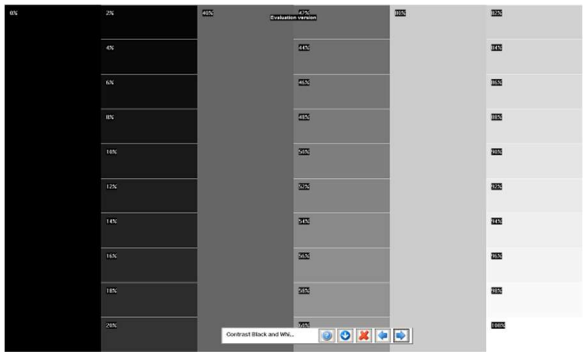
Listado de tests



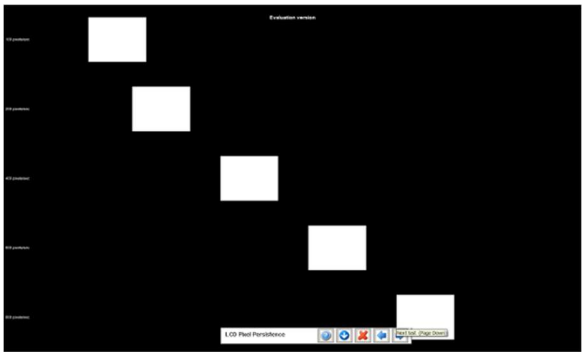
"Master Screen"



"Crosshatch"



"Contrast Black and White"

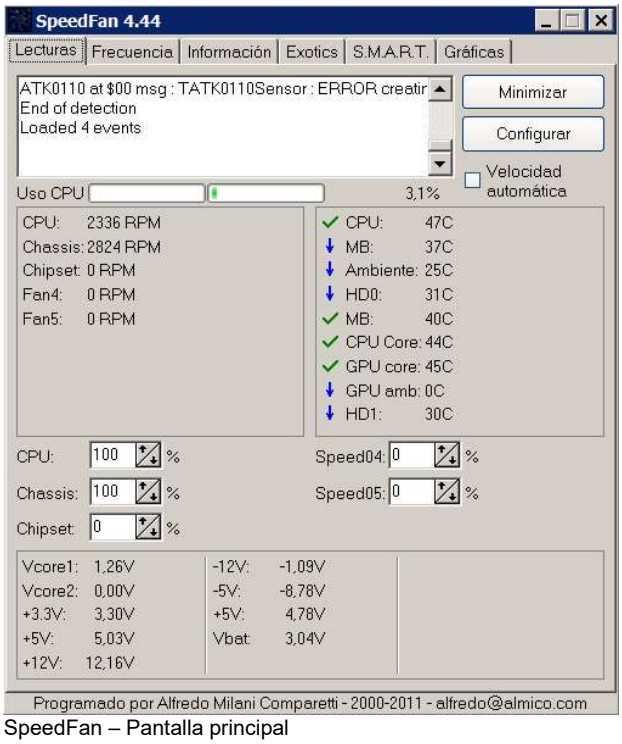


"LCD Pixel Persistence"

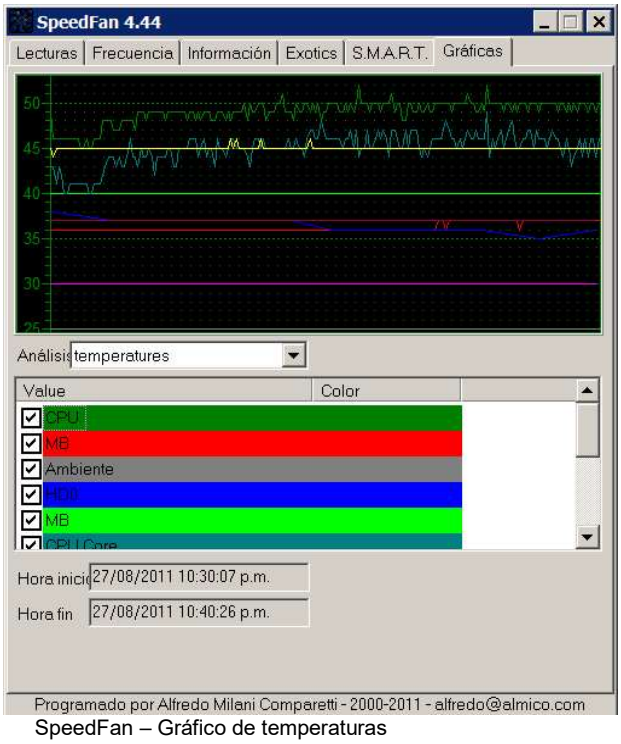
5.2 PROGRAMAS DE MONITOREO

5.2.1 Programas de monitoreo para Windows – Speedfan

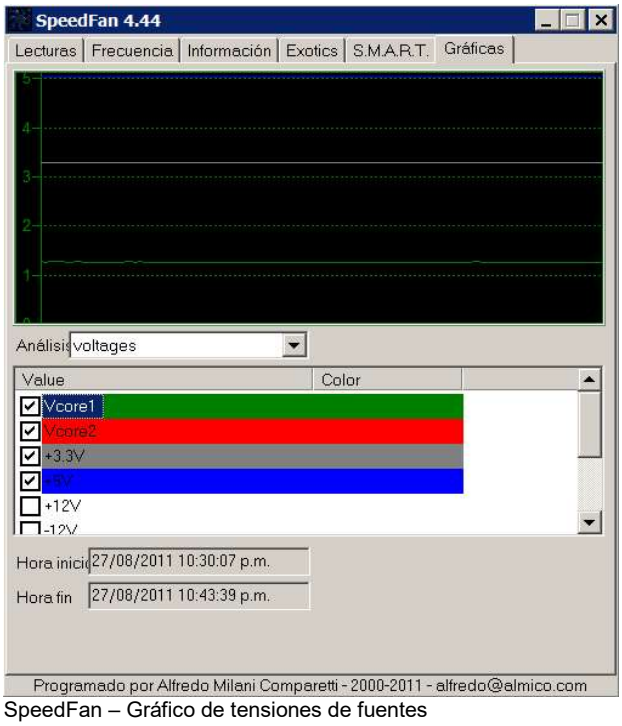
Este programa freeware fue concebido originalmente para controlar la velocidad de los ventiladores en función de la temperatura medida. Sin embargo, hoy es utilizado por muchos usuarios para monitorear las tensiones, velocidades de ventiladores y temperaturas en las placas que posean chip sensor. El programa es gratuito y la última versión puede descargarse del sitio del autor, Alfredo Milani Comparetti en: http://www.almico.com/speedfan.php.



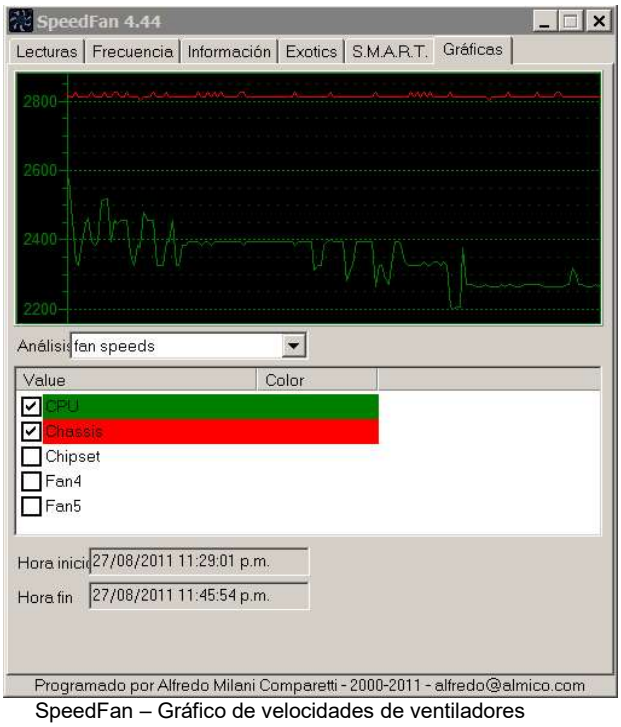
SpeedFan – Pantalla principal



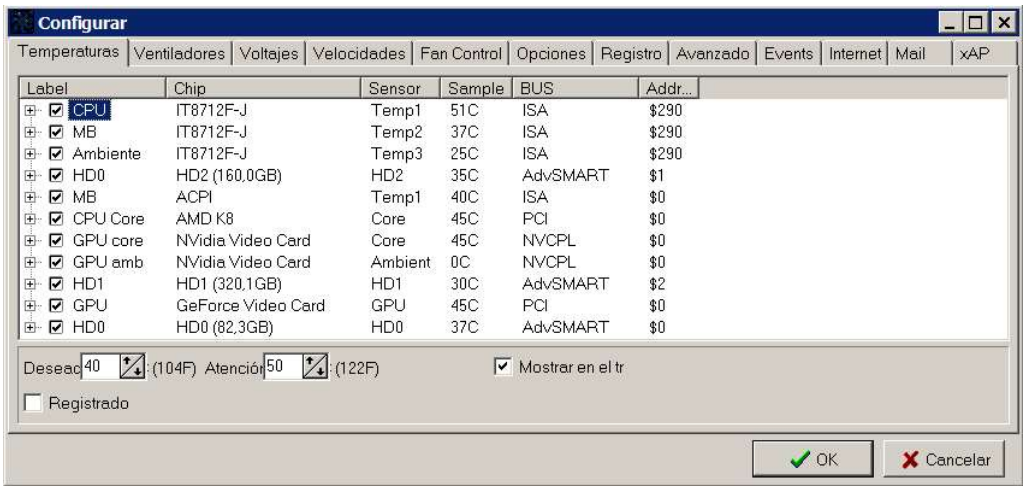
SpeedFan – Gráfico de temperaturas



SpeedFan – Gráfico de tensiones de fuentes

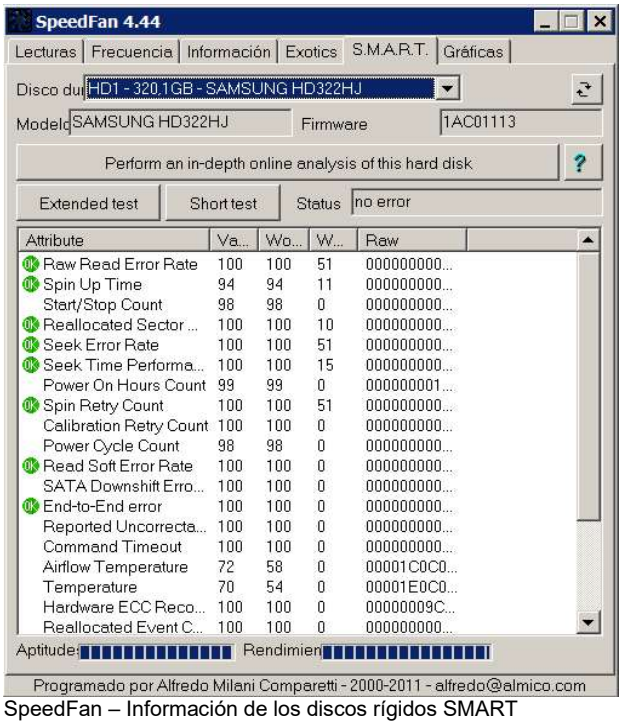


SpeedFan – Gráfico de velocidades de ventiladores



SpeedFan – Configuración de las mediciones de temperatura

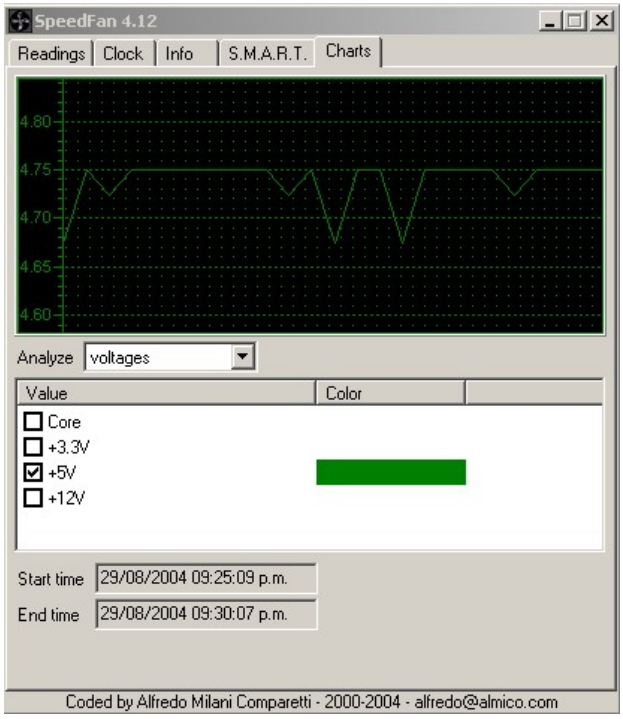
SpeedFan – Configuración de avisos ante eventos



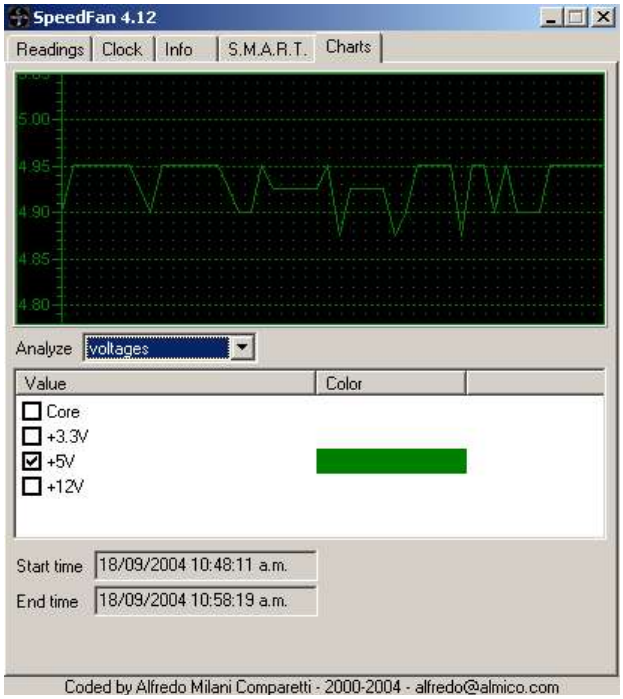
SpeedFan – Información de los discos rígidos SMART

Aplicación práctica de SpeedFan

En este gráfico se puede observar la variación de la tensión de la fuente de 5 V, 250 W cuando se requieren distintos consumos (rígido, CPU, video, etc.). Se puede observar que el valor base, con los consumos de reposo de los distintos componentes es de 4,75 V, en el límite de lo admisible por los dispositivos TTL.

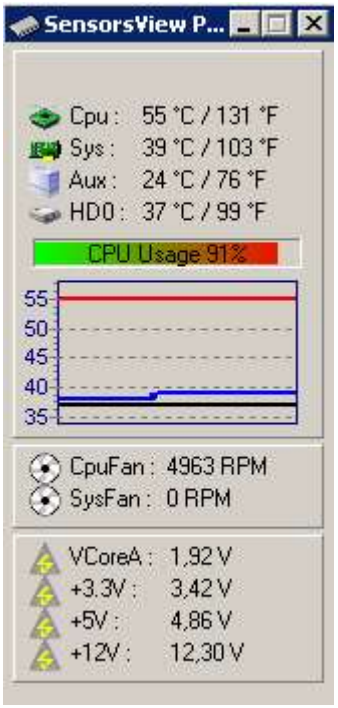


Cambiando la fuente de alimentación por una de 350 W se logró un valor adecuado.

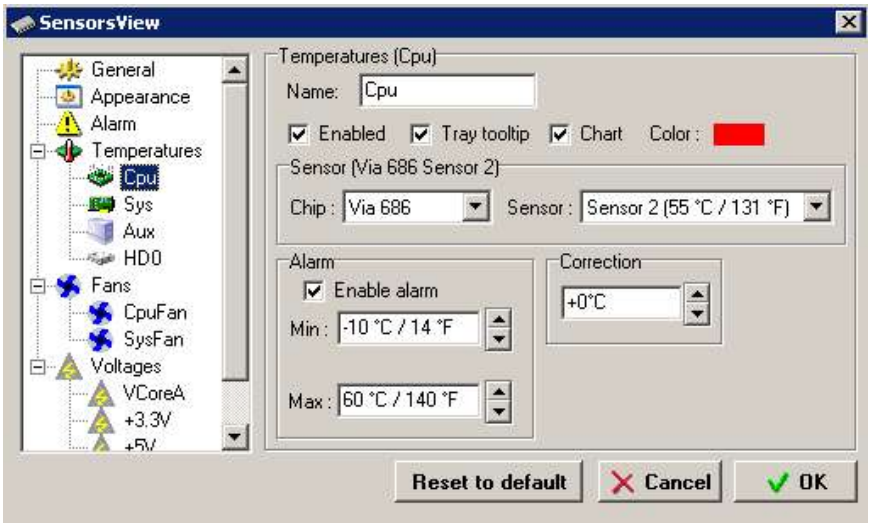


5.2.2 Programas de monitoreo para Windows – SensorsView

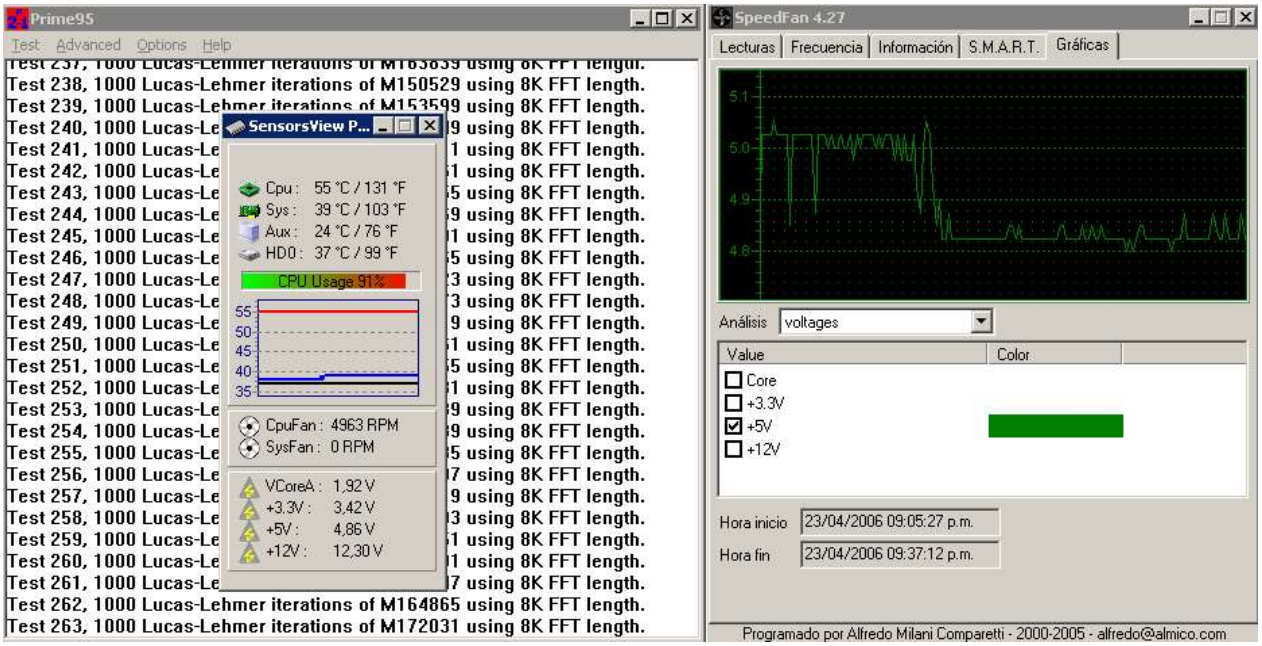
Un programa de utilidad similar al anterior es el SensorsView.



Aquí se puede ver la pantalla de configuración del programa, donde se pueden fijar valores de alarma, factores de corrección del valor indicado de temperatura, selección del chip sensor, etc.



El programa Prime95 puede ser usado para que la CPU trabaje a un régimen exhaustivo, permitiendo de este modo poner en evidencia fallas del procesador que ocurren sólo a altas temperaturas. Aquí se ven pantallas de SensorsView y Speedfan mientras se ejecuta Prime95. Véase la caída en la fuente de alimentación de 5 V.



5.2.3 Programas de monitoreo para Windows – Hardware Monitor

Hardware monitor es un programa de la empresa CPUID que detecta el chip de monitoreo de la PC y muestra los valores de tensiones, temperaturas, velocidades de los ventiladores de los distintos componentes (mother board, procesador, discos rígidos y placa de video). La última versión de este programa se puede descargar del sitio del fabricante en http://www.cpuid.com/.

La versión Pro, permite personalizar los rótulos, generar gráficos de tendencia y conectarse a otras PC de la red para realizar el monitoreo de estas.

CPUID Hardware Monitor

File Edit View Help

Sensor	Value	Min	Max
ATHLON4200			
ASUSTeK Computer INC....			
Voltages			
CPU VCORE	1.28 V	1.25 V	1.28 V
+3.3V	3.30 V	3.30 V	3.31 V
+5V	5.03 V	5.00 V	5.05 V
+12V	12.16 V	12.10 V	12.22 V
-12V	-12.35 V	-12.42 V	-12.29 V
+5V VCCH	4.78 V	4.78 V	4.78 V
VBAT	3.04 V	3.04 V	3.04 V
Temperatures			
THRM	40 °C (103 °F)	40 °C (103 °F)	40 °C (103 °F)
TMPIN0	46 °C (114 °F)	44 °C (111 °F)	50 °C (121 °F)
TMPIN1	36 °C (96 °F)	36 °C (96 °F)	36 °C (96 °F)
TMPIN2	25 °C (76 °F)	25 °C (76 °F)	25 °C (76 °F)
Fans			
FANIN0	2327 RPM	2257 RPM	2454 RPM
FANIN1	2800 RPM	2800 RPM	2824 RPM
Fans PWM			
FANPWM0	0 %	0 %	0 %
FANPWM1	0 %	0 %	0 %
FANPWM2	99 %	99 %	99 %
AMD Athlon 64 X2 4200+			
Temperatures			
Core #0	47 °C (116 °F)	46 °C (114 °F)	51 °C (123 °F)
Core #1	43 °C (109 °F)	40 °C (103 °F)	46 °C (114 °F)
HDS728080PLAT20			
Temperatures			
Assembly	38 °C (100 °F)	38 °C (100 °F)	38 °C (100 °F)
SAMSUNG HD160HJ			
Temperatures			
Assembly	37 °C (98 °F)	35 °C (94 °F)	37 °C (98 °F)
Air Flow	33 °C (91 °F)	31 °C (87 °F)	33 °C (91 °F)
SAMSUNG HD322HJ			
Temperatures			
Assembly	29 °C (84 °F)	29 °C (84 °F)	29 °C (84 °F)
Air Flow	26 °C (78 °F)	26 °C (78 °F)	26 °C (78 °F)
NVIDIA GeForce 8600 GT			
Temperatures			
TMPIN0	45 °C (112 °F)	44 °C (111 °F)	45 °C (112 °F)
TMPIN0	45 °C (112 °F)	44 °C (111 °F)	45 °C (112 °F)
Fans PWM			
FANPWMIN0	30 %	30 %	30 %

Ready

5.3 BENCHMARKS

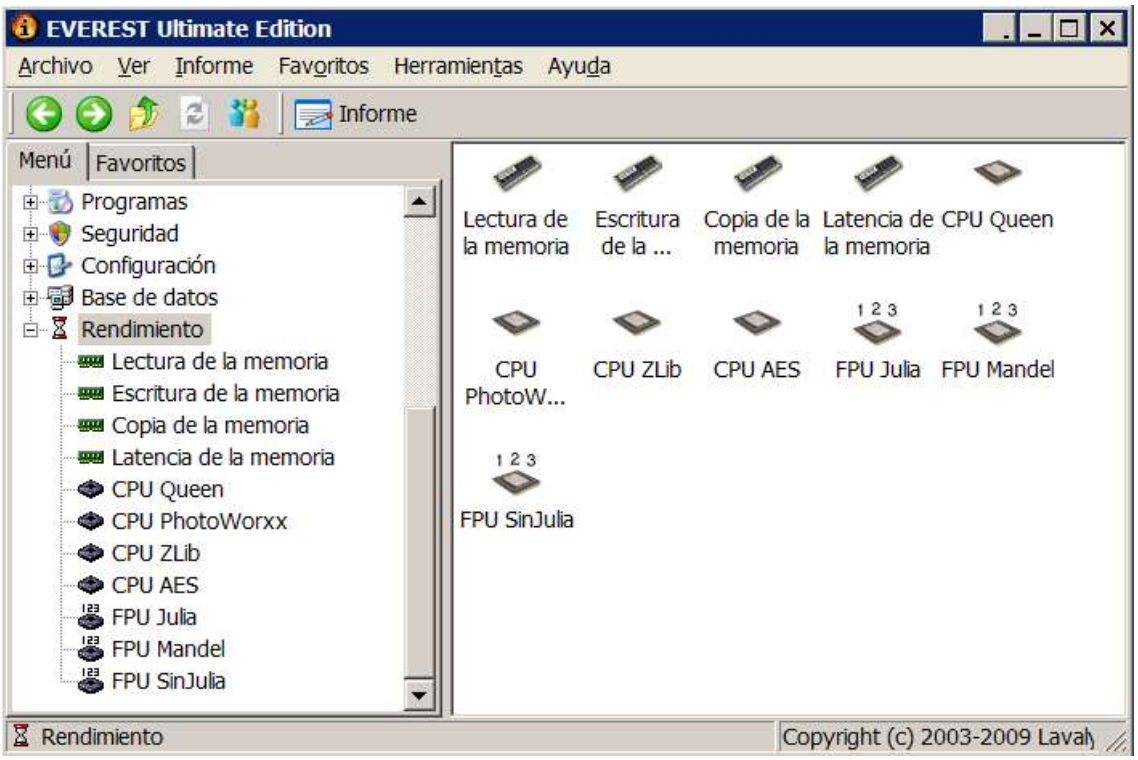
5.3.1 Introducción

Un benchmark es una comparativa de actividades similares. En el caso específico de los benchmarks de computadoras, son programas cuyo objetivo es determinar la performance de cada uno de sus componentes, y si fuera posible la performance total del sistema, a los efectos de comparar un sistema con otro. Esta información también aporta para generar las comparativas "precio/performance".

Si el mismo programa obtiene diferentes resultados para un determinado componente, por ejemplo, la velocidad de procesamiento aritmético del procesador A es mas alta que la del procesador B, esto debería significar que el procesador A se desempeñará mejor en una actividad real que haga uso intensivo del procesamiento aritmético. Para algunas aplicaciones especiales como juegos de video, existen benchmarks específicos para ese propósito, como el 3DMark, de Futuremark.

5.3.2 Benchmark para Windows – Everest

El programa Everest de la empresa Lavalys tiene su fortaleza en la información que brinda del sistema y cada uno de sus componentes, sin embargo, posee un apartado llamado "rendimiento" donde se pueden realizar diversos benchmarks. La última versión se puede descargar en forma gratuita de http://www.lavalys.com/.



EVEREST Ultimate Edition

Archivo Ver Informe Favoritos Herramientas Ayuda

Informe Resultados Parámetros

Menú Favoritos

EVEREST v5.30.1900
- Computadora
- Motherboard
- Sistema operativo
- Servidor
- Monitor
- Multimedia
- Almacenamiento
- Red
- DirectX
- Dispositivos de hardware
- Programas
- Seguridad
- Configuración
- Base de datos
- Rendimiento
  - Lectura de la memoria
  - Escritura de la memoria
  - Copia de la memoria
  - Latencia de la memoria
  - CPU Queen
  - CPU PhotoWorxx
  - CPU ZLib
  - CPU AES
  - FPU Julia
  - FPU Mandel
  - FPU SinJulia

CPU		Reloj de l...	Motherboard
17914	Xeon X3430	2400 MHz	Supermicro X8SI
16740	4x Xeon 5140	2333 MHz	Intel S5000VSA
16235	8x Opteron HE 2...	1700 MHz	Supermicro H8D
13694	4x Phenom X4 9...	2200 MHz	Asus M3A
12560	2x Core 2 Extre...	2933 MHz	Abit AB9
11182	2x Athlon64 X2 ...	3200 MHz	MSI K9N SLI Plati
10572	4x Opteron HE 2...	1800 MHz	Tyan Thunder h:
9574	2x Core 2 Duo P...	2266 MHz	MSI MegaBook P
7757	2x Core 2 Duo T...	1833 MHz	Asus F3000Jc Nc
7693	2x Core Duo T2...	2000 MHz	Asus N4L-VM DH
7267	2x Athlon64 X2 ...	2100 MHz	ASRock ALiveNF7
7221	**2x Athlon64 X...**	**2200 MHz**	**Asus M2N-E SI**
7098	2x Pentium EE 9...	3466 MHz	Intel D955XBK
6949	2x Xeon HT	3400 MHz	Intel SE7320SP2

Campo	Valor
Tipo de CPU	DualCore AMD Athlon 64 X2 4200+ (Winc
CPU Plataforma / Escalona...	Socket AM2 / BH-F2
Reloj de la CPU	2211.3 MHz (original: 2200 MHz)
Multiplicador de la CPU	11x
FSB de la CPU	201.0 MHz (original: 200 MHz)
Bus de la memoria	368.6 MHz
Relación DRAM:FSB	CPU/6
Chipset del motherboard	nVIDIA nForce 500 SLI, AMD Hammer

Queen 10-10: 0.61 sec (2 CPU, HT:off, mask:00000003) BDLL: 2.4.273.0, CPU: 2211 MHz, TSC:

EVEREST Ultimate Edition

Archivo Ver Informe Favoritos Herramientas Ayuda

Informe Resultados Parámetros

Menú Favoritos

EVEREST v5.30.1900
- Computadora
- Motherboard
- Sistema operativo
- Servidor
- Monitor
- Multimedia
- Almacenamiento
- Red
- DirectX
- Dispositivos de hardware
- Programas
- Seguridad
- Configuración
- Base de datos
- Rendimiento
  - Lectura de la memoria
  - Escritura de la memoria
  - Copia de la memoria
  - Latencia de la memoria
  - CPU Queen
  - CPU PhotoWorxx
  - CPU ZLib
  - CPU AES
  - FPU Julia
  - FPU Mandel
  - FPU SinJulia

CPU		Reloj de l...	Motherboard
7980	8x Opteron HE 2...	1700 MHz	Supermicro H8D
5517	2x Core 2 Extre...	2933 MHz	Abit AB9
5329	4x Phenom X4 9...	2200 MHz	Asus M3A
4192	2x Core 2 Duo P...	2266 MHz	MSI MegaBook P
3402	2x Core 2 Duo T...	1833 MHz	Asus F3000Jc Nc
3065	4x Opteron HE 2...	1800 MHz	Tyan Thunder h:
2711	2x Pentium EE 9...	3466 MHz	Intel D955XBK
2688	2x Athlon64 X2 ...	3200 MHz	MSI K9N SLI Plat
2638	2x Xeon HT	3400 MHz	Intel SE7320SP2
2072	2x Xeon	3066 MHz	Asus PCH-DL
1900	2x Pentium D 820	2800 MHz	Abit Fatal1ty F-I9
1774	2x Athlon64 X2 ...	2100 MHz	ASRock ALiveNF7
1749	**2x Athlon64 X...**	**2200 MHz**	**Asus M2N-E SI**
1625	2x Core Duo T2...	2000 MHz	Asus N4L-VM DH

Campo	Valor
Tipo de CPU	DualCore AMD Athlon 64 X2 4200+ (Winc
CPU Plataforma / Escalona...	Socket AM2 / BH-F2
Reloj de la CPU	2211.3 MHz (original: 2200 MHz)
Multiplicador de la CPU	11x
FSB de la CPU	201.0 MHz (original: 200 MHz)
Bus de la memoria	368.6 MHz
Relación DRAM:FSB	CPU/6
Chipset del motherboard	nVIDIA nForce 500 SLI, AMD Hammer

SSE2_Julia_4pix_New_2 (2 CPU, HT:off, mask:00000003) BDLL: 2.4.273.0, CPU: 2211 MHz, TSC:

EVEREST Ultimate Edition

Archivo Ver Informe Favoritos Herramientas Ayuda

Informe Resultados

Menú Favoritos

EVEREST v5.30.1900
- Computadora
- Motherboard
- Sistema operativo
- Servidor
- Monitor
- Multimedia
- Almacenamiento
- Red
- DirectX
- Dispositivos de hardware
- Programas
- Seguridad
- Configuración
- Base de datos
- Rendimiento
  - Lectura de la memoria
  - Escritura de la memoria
  - Copia de la memoria
  - Latencia de la memoria
  - CPU Queen
  - CPU PhotoWorxx
  - CPU ZLib
  - CPU AES
  - FPU Julia
  - FPU Mandel
  - FPU SinJulia

CPU		Reloj de la...	Motherboard
13937 MB/s	Core i7 Extreme ...	3333 MHz	Asus P6T Deluxe
12472 MB/s	Xeon X5550	2666 MHz	Supermicro X8D
11598 MB/s	Xeon X3430	2400 MHz	Supermicro X8S
8855 MB/s	Athlon64 X2 Blac...	3200 MHz	MSI K9N SLI Plat
8014 MB/s	Pentium EE 955	3466 MHz	Intel D955XBK
7893 MB/s	P4EE	3733 MHz	Intel SE7230NH
7659 MB/s	Phenom II X4 Bla...	3000 MHz	Asus M3N78-EM
7198 MB/s	Core 2 Extreme ...	3000 MHz	Gigabyte GA-EP
7039 MB/s	Core 2 Extreme ...	2933 MHz	Abit AB9
7003 MB/s	**Athlon64 X2 4...**	**2200 MHz**	**Asus M2N-E S**
6751 MB/s	Core 2 Extreme ...	2666 MHz	Intel D975XBX2
6349 MB/s	Athlon64 X2 400...	2100 MHz	ASRock ALiveNF
6203 MB/s	Pentium D 820	2800 MHz	Abit Fatal1ty F-I
5961 MB/s	Athlon64 3200+	2000 MHz	ASRock 939S56

Campo	Valor
Tipo de CPU	DualCore AMD Athlon 64 X2 4200+ (Winc
CPU Plataforma / Escalona...	Socket AM2 / BH-F2
Reloj de la CPU	2211.3 MHz (original: 2200 MHz)
Multiplicador de la CPU	11x
FSB de la CPU	201.0 MHz (original: 200 MHz)
Bus de la memoria	368.6 MHz
Relación DRAM:FSB	CPU/6
Chipset del motherboard	nVIDIA nForce 500 SLI, AMD Hammer

Load Mem MMX+ PrefNTA DEC: 3.17 bytes/cycle (mask:00000001) | BDLL: 2.4.273.0, CPU: 2211 MHz, TSC:

EVEREST Ultimate Edition

Archivo Ver Informe Favoritos Herramientas Ayuda

Informe Resultados Parámetros

Menú Favoritos

EVEREST v5.30.1900
- Computadora
- Motherboard
- Sistema operativo
- Servidor
- Monitor
- Multimedia
- Almacenamiento
- Red
- DirectX
- Dispositivos de hardware
- Programas
- Seguridad
- Configuración
- Base de datos
- Rendimiento
  - Lectura de la memoria
  - Escritura de la memoria
  - Copia de la memoria
  - Latencia de la memoria
  - CPU Queen
  - CPU PhotoWorxx
  - CPU ZLib
  - CPU AES
  - FPU Julia
  - FPU Mandel
  - FPU SinJulia

CPU		Reloj de l...	Motherboard
79105 KB/s	4x Phenom II X4 ...	3000 MHz	Asus M3N78-EM
77575 KB/s	4x Core 2 Extre...	3000 MHz	Gigabyte GA-EP.
70530 KB/s	4x Core 2 Extre...	2666 MHz	Intel D975XBX2
61460 KB/s	4x Xeon 5140	2333 MHz	Intel S5000VSA
58747 KB/s	4x Phenom X4 9...	2200 MHz	Asus M3A
58577 KB/s	Xeon X3430	2400 MHz	Supermicro X8S.
42550 KB/s	4x Opteron HE 2...	1800 MHz	Tyan Thunder h.
39173 KB/s	2x Core 2 Extre...	2933 MHz	Abit AB9
38029 KB/s	2x Athlon64 X2 B...	3200 MHz	MSI K9N SLI Plat
29787 KB/s	2x Pentium EE 9...	3466 MHz	Intel D955XBK
29375 KB/s	2x Core 2 Duo P...	2266 MHz	MSI MegaBook ..
29210 KB/s	2x Xeon HT	3400 MHz	Intel SE7320SP2
24965 KB/s	2x Xeon	3066 MHz	Asus PCH-DL
24280 KB/s	**2x Athlon64 X...**	**2200 MHz**	**Asus M2N-E SI**

Campo	Valor
Tipo de CPU	DualCore AMD Athlon 64 X2 4200+ (Wind
CPU Plataforma / Escalona...	Socket AM2 / BH-F2
Reloj de la CPU	2211.3 MHz (original: 2200 MHz)
Multiplicador de la CPU	11x
FSB de la CPU	201.0 MHz (original: 200 MHz)
Bus de la memoria	368.5 MHz
Relación DRAM:FSB	CPU/6
Chipset del motherboard	nVIDIA nForce 500 SLI, AMD Hammer

Pack 32768 KB: 2.98 sec (2 CPU, HT:off, mask:00000003) | BDLL: 2.4.273.0, CPU: 2211 MHz, TSC:

5.3.3 Benchmark para Windows – Sisoft Sandra

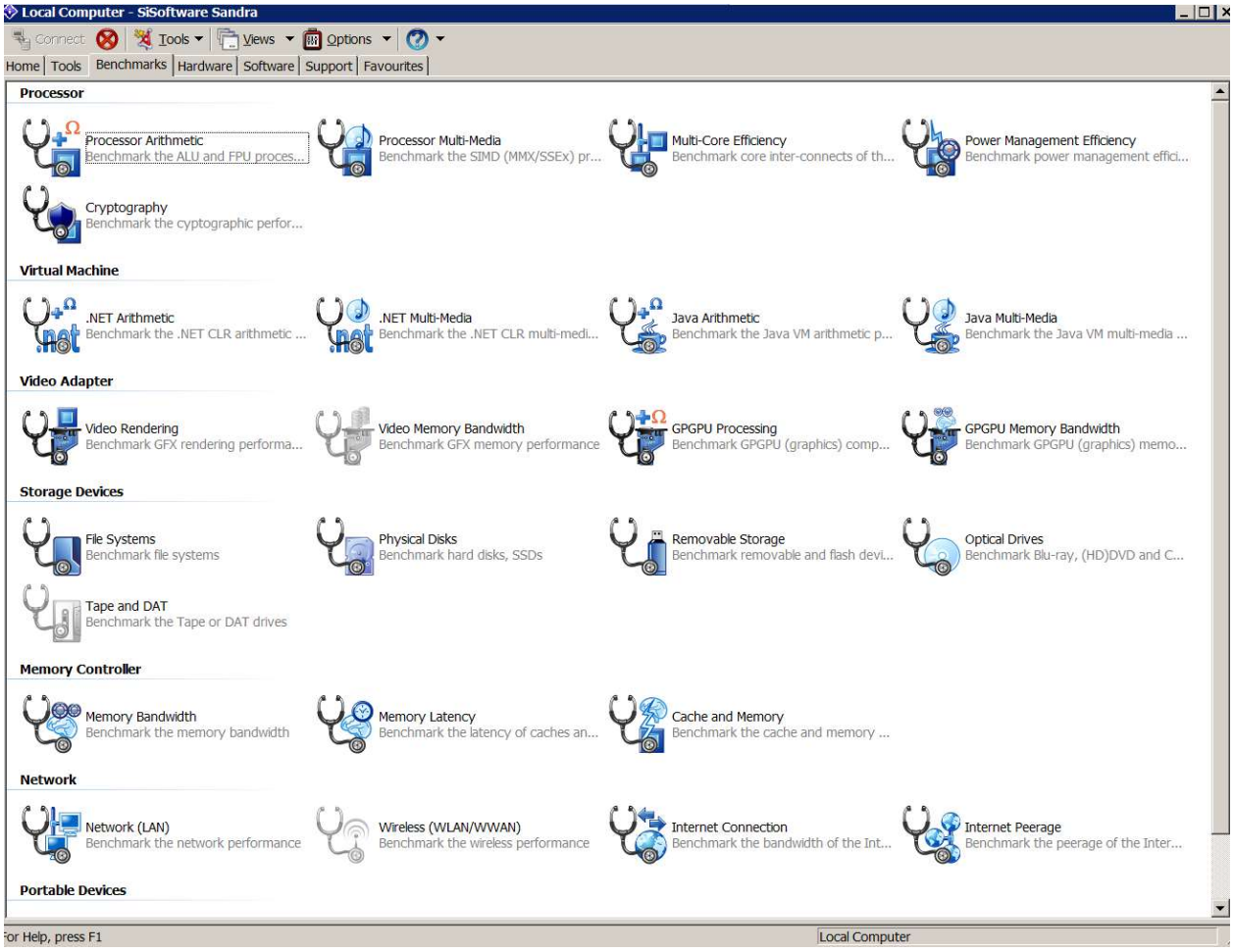
A continuación, se verán algunas características de un conocido programa, el Sandra 2009 de la empresa Sisoft. La última versión se puede descargar de http://www.sisoftware.net/.

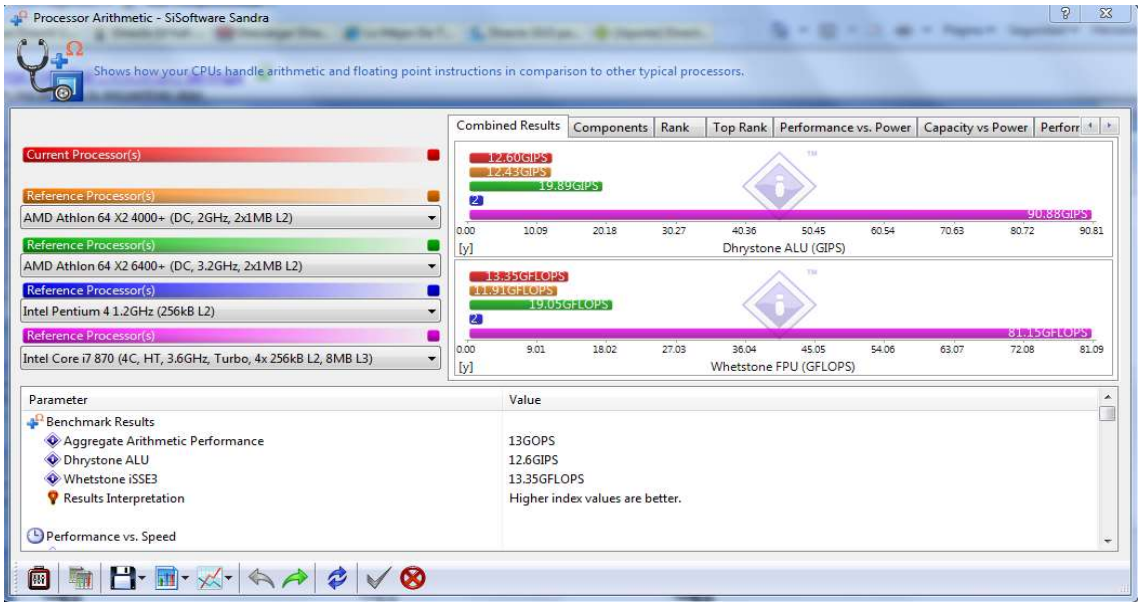
Según el fabricante, SiSoftware Sandra (the System ANalyser, Diagnostic and Reporting Assistant) es una utilidad de información y diagnóstico. Le proveerá gran cantidad de información (inclusive no documentada) que necesita conocer sobre su computadora, ya sea hardware o software.

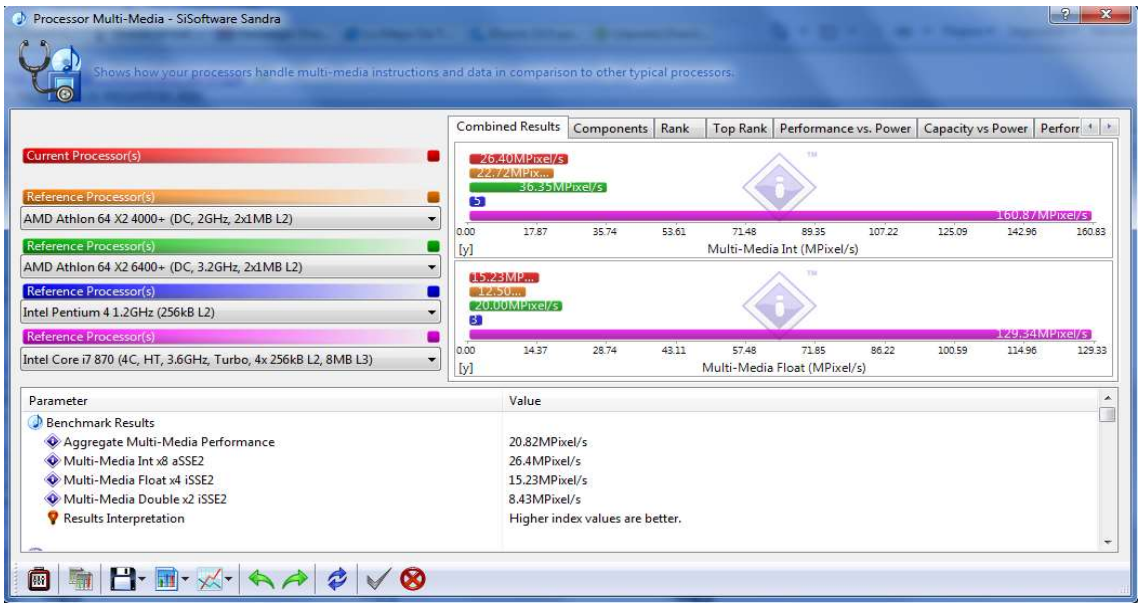
Funciona de la misma forma que otras utlidades de Windows, no obstante, intenta ir mas allá y mostrarle lo que realmente está sucediendo. Brinda al usuario la posibilidad de dibujar comparativas, tanto de alto como de bajo nivel. Puede conseguir información de la CPU, chipset, adaptador de video, puertos, impresoras, placa de sonido, memoria, red, "internals" de Windows, AGP, PCI, PCI-X, PCIe (PCI Express), USB, USB2, 1394/Firewire, y mas.

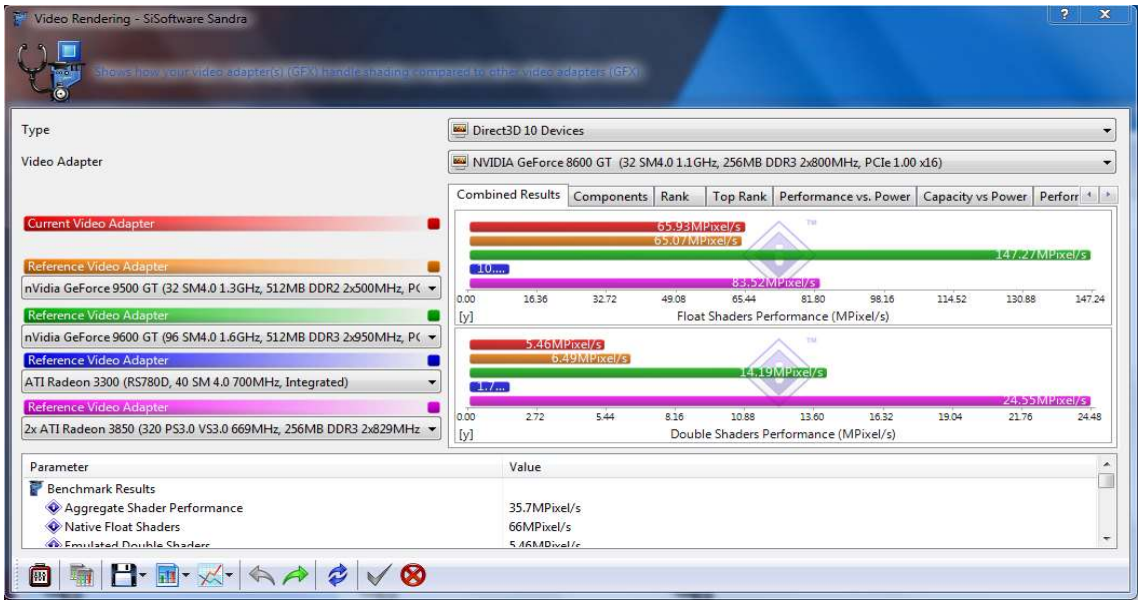
El equipo bajo test tiene las siguientes características:

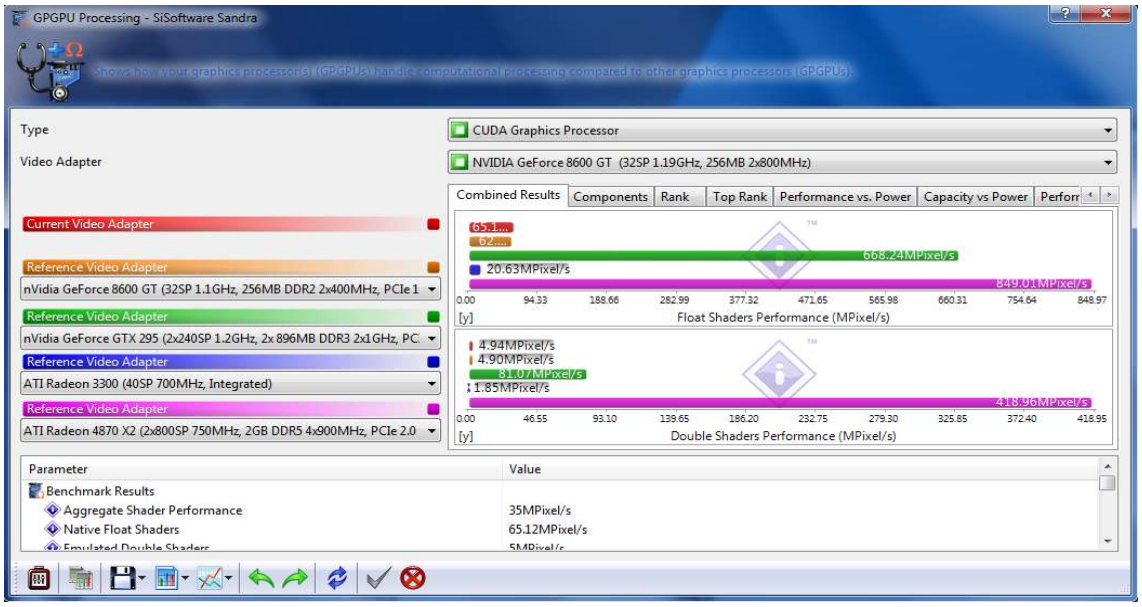
Procesador AMD Athlon 64 4200 @ 2200 MHz, zócalo AM2, 1,25V, cache L1 64 kB código y 64 kB datos, cache L2 512 kB, Mother ASUS M2NE-SLI, chipset Nforce500, FSB 200 MHz, SDRAM DDR2-800 de 3 GB (dual channel), bus PCI-E 2,5 GHz, bus PCI 33 MHz, bus USB 48 MHz, Video MSI Geforce 8600 GT de 256 MB PCI-E x 16, CD/DVD RW 32/8x UDMA 133, disco rígido Samsung 320 GB SATA-2 300 MB/s. Windows 7.

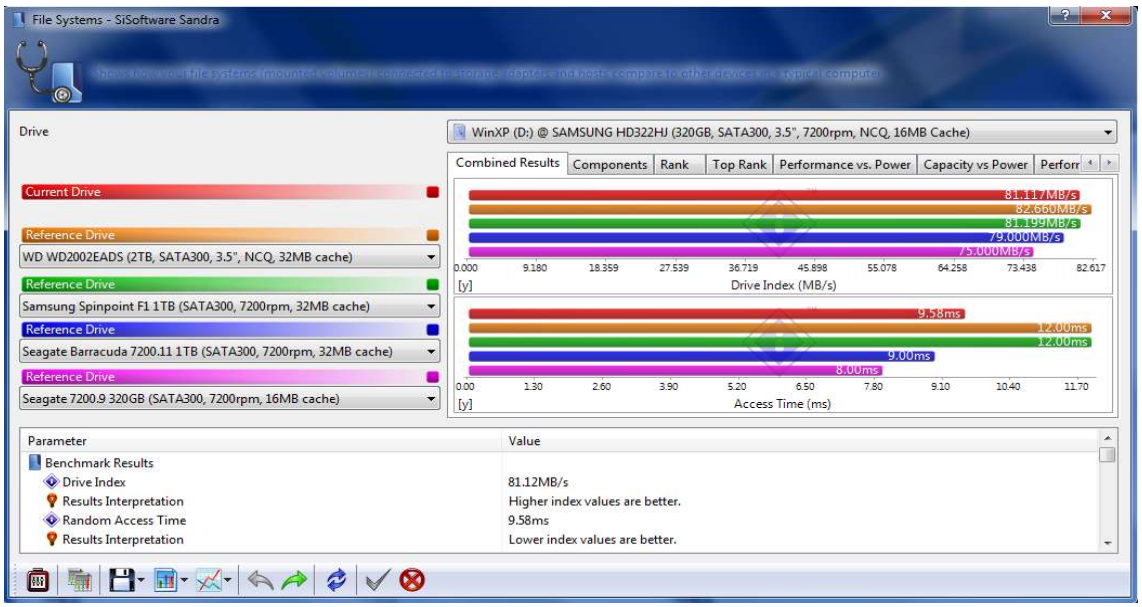


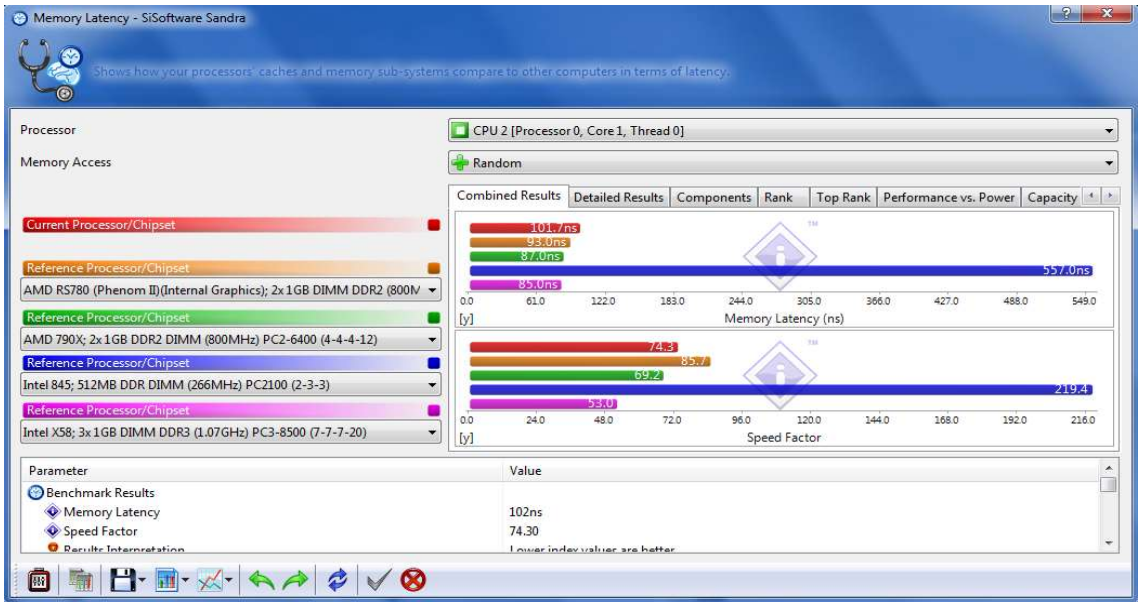

Processor Arithmetic - SiSoftware Sandra
Shows how your CPUs handle arithmetic and floating point instructions in comparison to other typical processors.
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Processor(s)
Reference Processor(s)
AMD Athlon 64 X2 4000+ (DC, 2GHz, 2x1MB L2)
Reference Processor(s)
AMD Athlon 64 X2 6400+ (DC, 3.2GHz, 2x1MB L2)
Reference Processor(s)
Intel Pentium 4 1.2GHz (256kB L2)
Reference Processor(s)
Intel Core i7 870 (4C, HT, 3.6GHz, Turbo, 4x 256kB L2, 8MB L3)
12.60GIPS
12.43GIPS
19.89GIPS
90.88GIPS
Dhrystone ALU (GIPS)
13.35GFLOPS
19.05GFLOPS
81.15GFLOPS
Whetstone FPU (GFLOPS)
Parameter
Value
Benchmark Results
Aggregate Arithmetic Performance
13GOPS
Dhrystone ALU
12.6GIPS
Whetstone iSSE3
13.35GFLOPS
Results Interpretation
Higher index values are better.
Performance vs. Speed

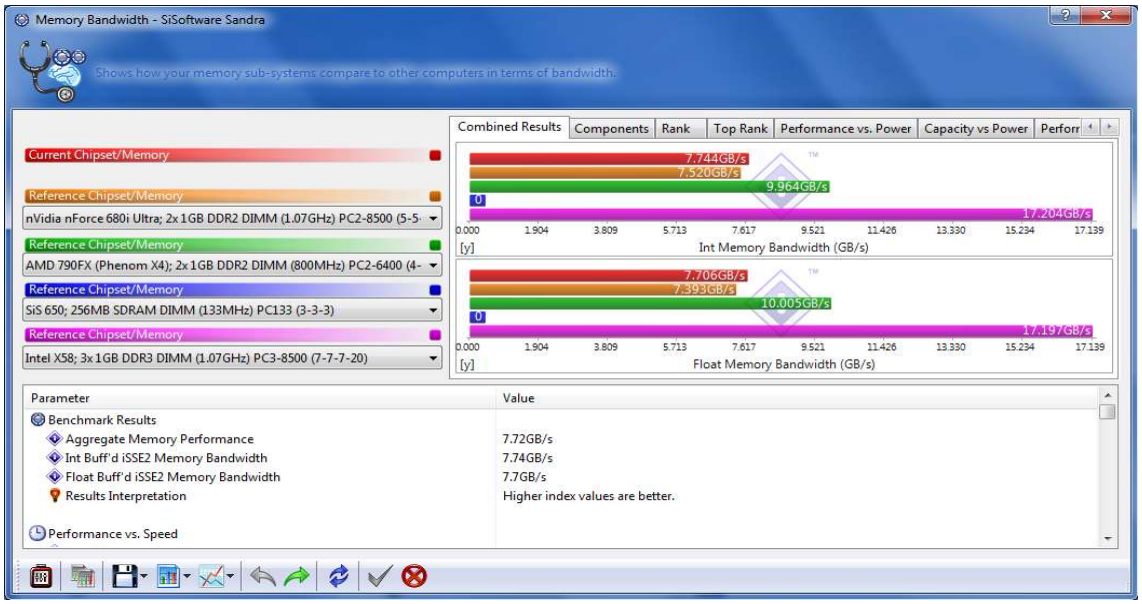

Processor Multi-Media - SiSoftware Sandra
Shows how your processors handle multi-media instructions and data in comparison to other typical processors.
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Processor(s)
Reference Processor(s)
AMD Athlon 64 X2 4000+ (DC, 2GHz, 2x1MB L2)
Reference Processor(s)
AMD Athlon 64 X2 6400+ (DC, 3.2GHz, 2x1MB L2)
Reference Processor(s)
Intel Pentium 4 1.2GHz (256kB L2)
Reference Processor(s)
Intel Core i7 870 (4C, HT, 3.6GHz, Turbo, 4x 256kB L2, 8MB L3)
26.40MPixel/s
36.35MPixel/s
160.87MPixel/s
Multi-Media Int (MPixel/s)
20.00MPixel/s
129.34MPixel/s
Multi-Media Float (MPixel/s)
Parameter
Value
Benchmark Results
Aggregate Multi-Media Performance
20.82MPixel/s
Multi-Media Int x8 aSSE2
26.4MPixel/s
Multi-Media Float x4 iSSE2
15.23MPixel/s
Multi-Media Double x2 iSSE2
8.43MPixel/s
Results Interpretation
Higher index values are better.

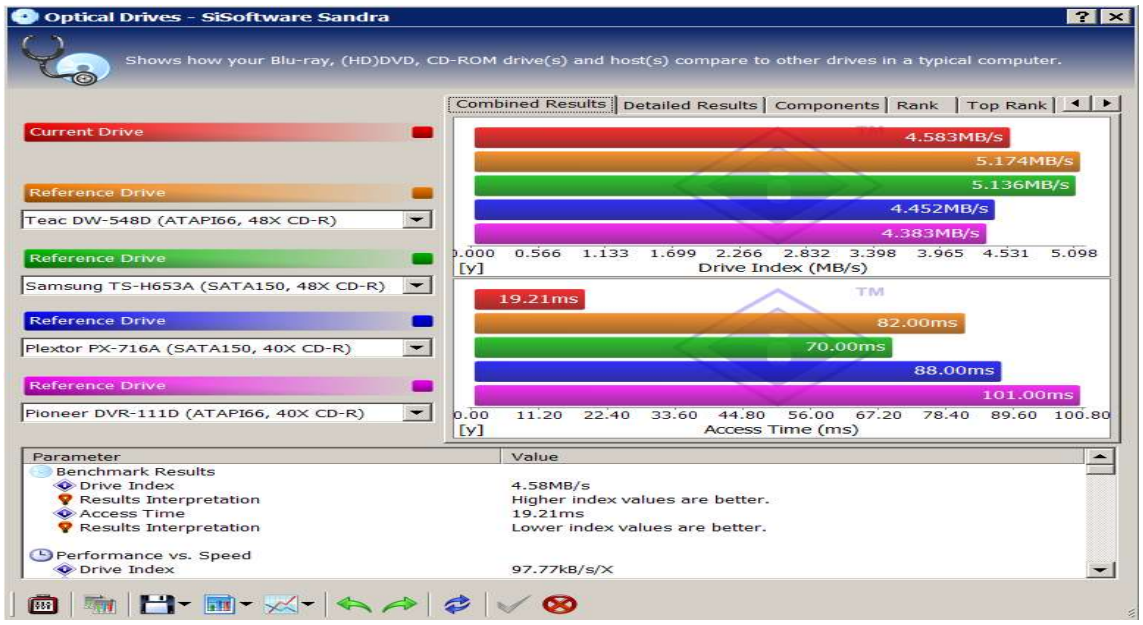

Video Rendering - SiSoftware Sandra
Type
Direct3D 10 Devices
Video Adapter
NVIDIA GeForce 8600 GT (32 SM4.0 1.1GHz, 256MB DDR3 2x800MHz, PCIe 1.00 x16)
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Video Adapter
Reference Video Adapter
nVidia GeForce 9500 GT (32 SM4.0 1.3GHz, 512MB DDR2 2x500MHz, P(
Reference Video Adapter
nVidia GeForce 9600 GT (96 SM4.0 1.6GHz, 512MB DDR3 2x950MHz, P(
Reference Video Adapter
ATI Radeon 3300 (RS780D, 40 SM 4.0 700MHz, Integrated)
Reference Video Adapter
2x ATI Radeon 3850 (320 PS3.0 VS3.0 669MHz, 256MB DDR3 2x829MHz
65.93MPixel/s
65.07MPixel/s
147.27MPixel/s
Float Shaders Performance (MPixel/s)
5.46MPixel/s
6.49MPixel/s
24.19MPixel/s
24.55MPixel/s
Double Shaders Performance (MPixel/s)
Parameter
Value
Benchmark Results
Aggregate Shader Performance
35.7MPixel/s
Native Float Shaders
66MPixel/s

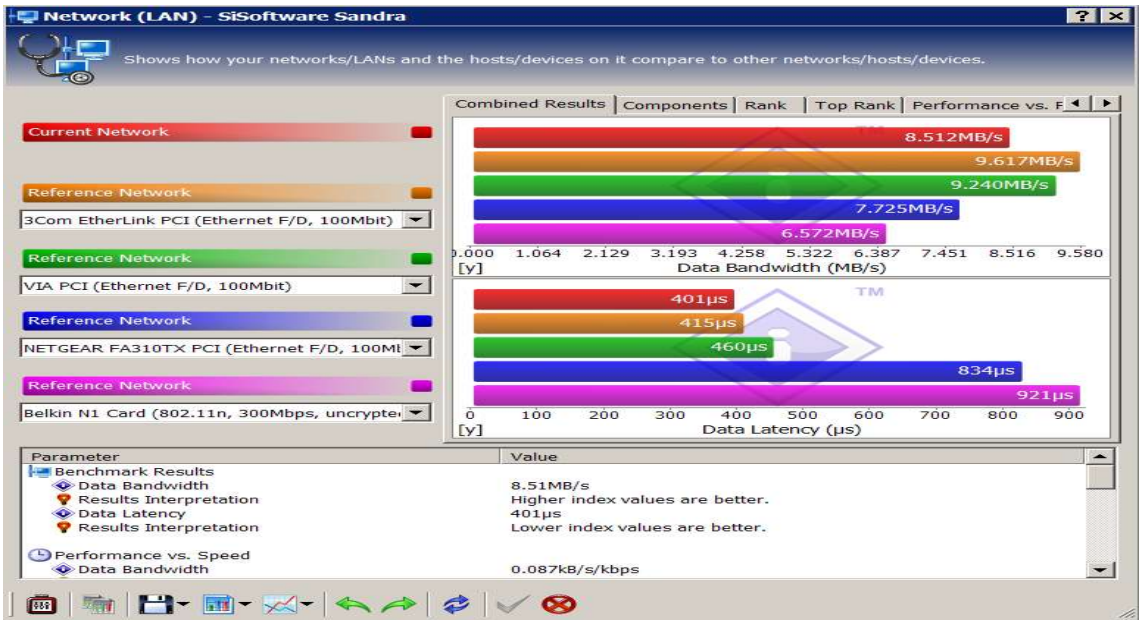

GPGPU Processing - SiSoftware Sandra
Type
CUDA Graphics Processor
Video Adapter
NVIDIA GeForce 8600 GT (32SP 1.19GHz, 256MB 2x800MHz)
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Video Adapter
Reference Video Adapter
nVidia GeForce 8600 GT (32SP 1.1GHz, 256MB DDR2 2x400MHz, PCIe 1
Reference Video Adapter
nVidia GeForce GTX 295 (2x240SP 1.2GHz, 2x 896MB DDR3 2x1GHz, PC
Reference Video Adapter
ATI Radeon 3300 (40SP 700MHz, Integrated)
Reference Video Adapter
ATI Radeon 4870 X2 (2x800SP 750MHz, 2GB DDR5 4x900MHz, PCIe 2.0
20.63MPixel/s
668.24MPixel/s
849.01MPixel/s
Float Shaders Performance (MPixel/s)
4.94MPixel/s
4.90MPixel/s
81.07MPixel/s
1.85MPixel/s
418.96MPixel/s
Double Shaders Performance (MPixel/s)
Parameter
Value
Benchmark Results
Aggregate Shader Performance
35MPixel/s
Native Float Shaders
65.12MPixel/s


File Systems - SiSoftware Sandra
Drive
WinXP (D:) @ SAMSUNG HD322HJ (320GB, SATA300, 3.5", 7200rpm, NCQ, 16MB Cache)
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Drive
Reference Drive
WD WD2002EADS (2TB, SATA300, 3.5", NCQ, 32MB cache)
Reference Drive
Samsung Spinpoint F1 1TB (SATA300, 7200rpm, 32MB cache)
Reference Drive
Seagate Barracuda 7200.11 1TB (SATA300, 7200rpm, 32MB cache)
Reference Drive
Seagate 7200.9 320GB (SATA300, 7200rpm, 16MB cache)
81.117MB/s
82.660MB/s
81.199MB/s
79.000MB/s
75.000MB/s
Drive Index (MB/s)
9.58ms
12.00ms
12.00ms
9.00ms
8.00ms
Access Time (ms)
Parameter
Value
Benchmark Results
Drive Index
81.12MB/s
Results Interpretation
Higher index values are better.
Random Access Time
9.58ms
Results Interpretation
Lower index values are better.


Memory Latency - SiSoftware Sandra
Processor
CPU 2 [Processor 0, Core 1, Thread 0]
Memory Access
Random
Combined Results
Detailed Results
Components
Rank
Top Rank
Performance vs. Power
Capacity
Current Processor/Chipset
Reference Processor/Chipset
AMD RS780 (Phenom II)(Internal Graphics); 2x 1GB DIMM DDR2 (800M
Reference Processor/Chipset
AMD 790X; 2x 1GB DDR2 DIMM (800MHz) PC2-6400 (4-4-4-12)
Reference Processor/Chipset
Intel 845; 512MB DDR DIMM (266MHz) PC2100 (2-3-3)
Reference Processor/Chipset
Intel X58; 3x 1GB DIMM DDR3 (1.07GHz) PC3-8500 (7-7-7-20)
101.7ns
93.0ns
87.0ns
557.0ns
85.0ns
Memory Latency (ns)
74.3
85.7
69.2
219.4
53.0
Speed Factor
Parameter
Value
Benchmark Results
Memory Latency
102ns
Speed Factor
74.30

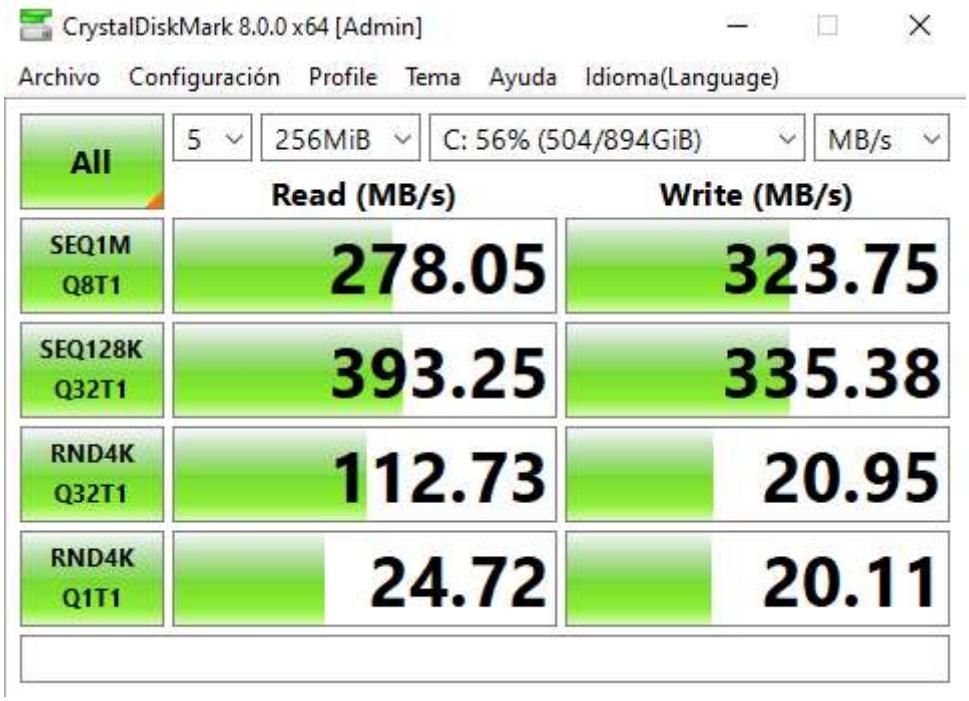

Memory Bandwidth - SiSoftware Sandra
Shows how your memory sub-systems compare to other computers in terms of bandwidth.
Combined Results
Components
Rank
Top Rank
Performance vs. Power
Capacity vs Power
Current Chipset/Memory
Reference Chipset/Memory
nVidia nForce 680i Ultra; 2x1GB DDR2 DIMM (1.07GHz) PC2-8500 (5-5-
Reference Chipset/Memory
AMD 790FX (Phenom X4); 2x1GB DDR2 DIMM (800MHz) PC2-6400 (4-
Reference Chipset/Memory
SiS 650; 256MB SDRAM DIMM (133MHz) PC133 (3-3-3)
Reference Chipset/Memory
Intel X58; 3x1GB DDR3 DIMM (1.07GHz) PC3-8500 (7-7-7-20)
7.744GB/s
7.520GB/s
9.964GB/s
17.204GB/s
Int Memory Bandwidth (GB/s)
7.706GB/s
7.393GB/s
10.005GB/s
17.197GB/s
Float Memory Bandwidth (GB/s)
Parameter
Value
Benchmark Results
Aggregate Memory Performance
7.72GB/s
Int Buff'd iSSE2 Memory Bandwidth
7.74GB/s
Float Buff'd iSSE2 Memory Bandwidth
7.7GB/s
Results Interpretation
Higher index values are better.
Performance vs. Speed


Optical Drives - SiSoftware Sandra
Shows how your Blu-ray, (HD)DVD, CD-ROM drive(s) and host(s) compare to other drives in a typical computer.
Combined Results
Detailed Results
Components
Rank
Top Rank
Current Drive
Reference Drive
Teac DW-548D (ATAPI66, 48X CD-R)
Reference Drive
Samsung TS-H653A (SATA150, 48X CD-R)
Reference Drive
Plextor PX-716A (SATA150, 40X CD-R)
Reference Drive
Pioneer DVR-111D (ATAPI66, 40X CD-R)
4.583MB/s
5.174MB/s
5.136MB/s
4.452MB/s
4.383MB/s
Drive Index (MB/s)
19.21ms
82.00ms
70.00ms
88.00ms
101.00ms
Access Time (ms)
Parameter
Value
Benchmark Results
Drive Index
4.58MB/s
Results Interpretation
Higher index values are better.
Access Time
19.21ms
Results Interpretation
Lower index values are better.
Performance vs. Speed
Drive Index
97.77kB/s/X


Network (LAN) - SiSoftware Sandra
Shows how your networks/LANs and the hosts/devices on it compare to other networks/hosts/devices.
Combined Results
Components
Rank
Top Rank
Current Network
Reference Network
3Com EtherLink PCI (Ethernet F/D, 100Mbit)
Reference Network
VIA PCI (Ethernet F/D, 100Mbit)
Reference Network
NETGEAR FA310TX PCI (Ethernet F/D, 100Mb
Reference Network
Belkin N1 Card (802.11n, 300Mbps, uncrypte
8.512MB/s
9.617MB/s
9.240MB/s
7.725MB/s
6.572MB/s
Data Bandwidth (MB/s)
401µs
415µs
460µs
834µs
921µs
Data Latency (µs)
Parameter
Value
Benchmark Results
Data Bandwidth
8.51MB/s
Results Interpretation
Higher index values are better.
Data Latency
401µs
Results Interpretation
Lower index values are better.
Performance vs. Speed
Data Bandwidth
0.087kB/s/kbps

5.3.4 Benchmark específico para discos en Windows – CrystalDiskMark

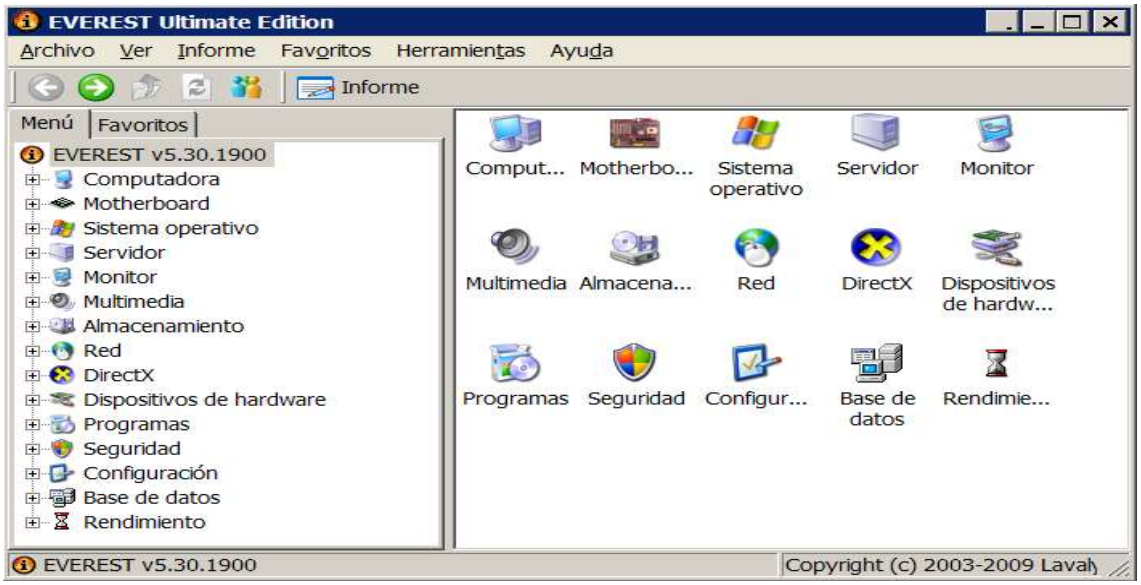
Este programa es gratuito y se puede descargar del sitio de su autor en:
https://crystalmark.info/en/software/crystaldiskmark/.
La versión más reciente es la 8.0.4, de 2021.

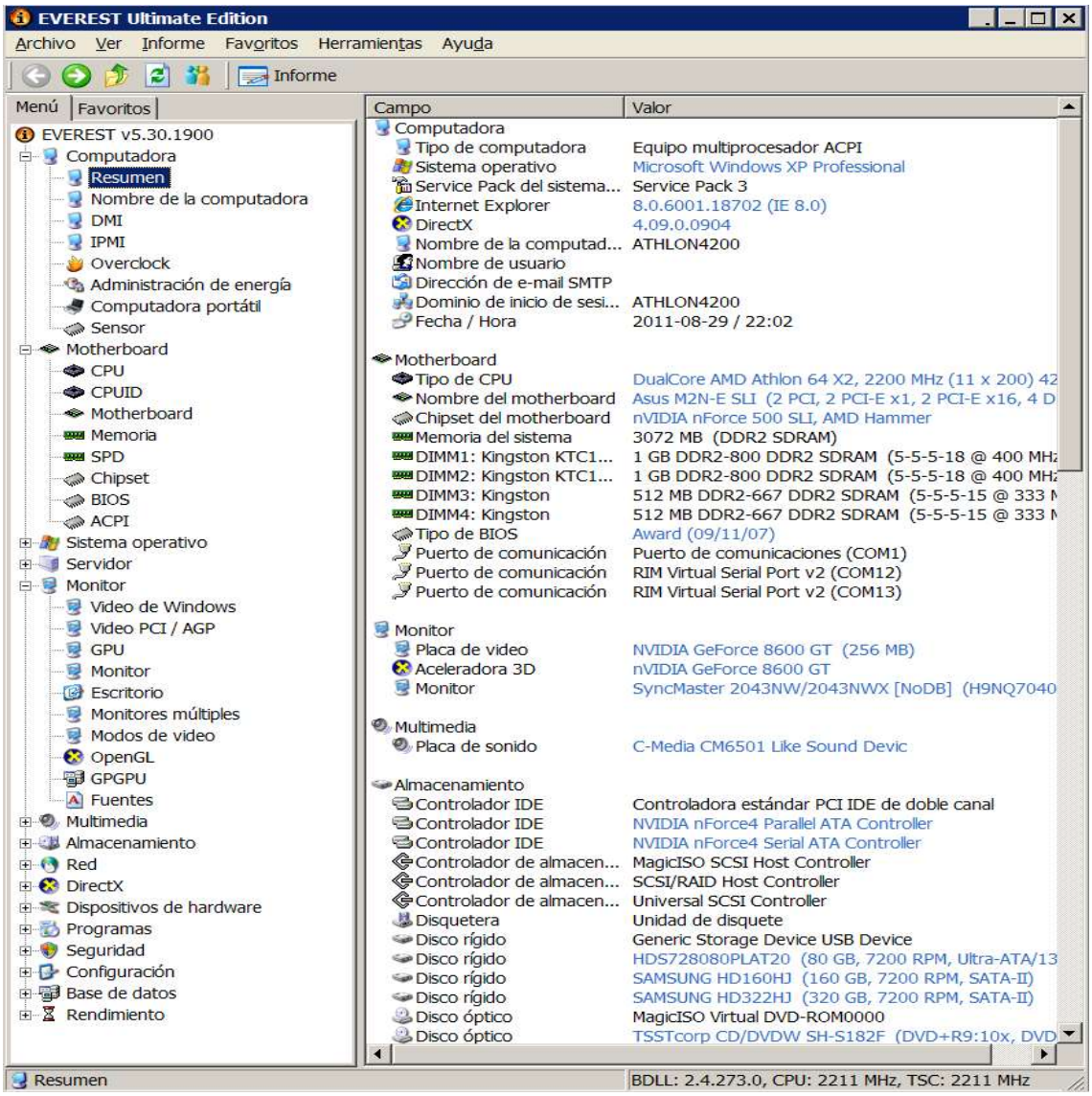


En la imagen se ven valores correspondientes a una unidad SSD

5.4 PROGRAMAS DE INFORMACIÓN

5.4.1 Programas de información para Windows – Everest





CAPÍTULO 6

SEGURIDAD INFORMÁTICA

6.1 DEFINICIONES

SEGURIDAD INFORMÁTICA

Podemos definir la **seguridad informática** o **ciberseguridad** como el proceso de prevenir y detectar el uso no autorizado de un sistema informático. Implica las actividades de proteger contra intrusos el uso de nuestros recursos informáticos, especialmente la información contenida en una computadora o circulante a través de las redes de computadoras, con intenciones maliciosas, con intención de obtener ganancias o incluso la posibilidad de acceder a ellos por accidente.

La seguridad informática abarca una serie de medidas de seguridad, tales como programas de software de antivirus, firewalls y otras medidas que dependen del usuario, tales como la activación de la desactivación de ciertas funciones de software, como scripts de Java, ActiveX, cuidar del uso adecuado de la computadora, los recursos de red o de Internet.

MALWARE

El malware (una contracción para soft**ware mal**icioso) es cualquier software diseñado intencionalmente para causar daños a una computadora, servidor, cliente o red informática. Por el contrario, el software que causa daños involuntarios debido a alguna deficiencia se describe típicamente como un error de software (bug). Existe una amplia variedad de tipos de malware, incluyendo virus informáticos, gusanos, troyanos, ransomware, spyware, adware, wiper y scareware, entre otros.

VIRUS INFORMÁTICO

Un **virus informático** es un tipo de programa informático que, cuando se ejecuta, se replica modificando otros programas informáticos e insertando su propio código. Si esta replicación tiene éxito, se dice que las áreas afectadas están "infectadas" con un virus informático, una analogía derivada de los virus biológicos.

Los virus informáticos generalmente requieren un programa host. El virus escribe su propio código en el programa anfitrión. Cuando se ejecuta el programa, el programa de virus escrito se ejecuta primero, causando infección y daños. Un gusano informático no necesita un programa host, ya que es un programa independiente o fragmento de código. Por lo tanto, no está restringido por el programa anfitrión, pero puede ejecutarse de forma independiente y llevar a cabo ataques de forma activa.

6.2 ASPECTOS QUE ATAÑEN A LA SEGURIDAD INFORMÁTICA

Las cuatro áreas principales que cubre la seguridad informática son:

1. **Confidencialidad**: Sólo los usuarios autorizados pueden acceder a nuestros recursos, datos e información.
2. **Integridad**: Sólo los usuarios autorizados deben ser capaces de modificar los datos cuando sea necesario.

3. **Disponibilidad**: Los datos deben estar disponibles para los usuarios cuando sea necesario.
4. **Autenticación**: Estás realmente comunicándote con los que piensas que te estás comunicando.

6.2.1 Medidas de mantenimiento y prevención de intrusiones

Los ataques más utilizados en contra de un sistema informático son los troyanos, los gusanos y la suplantación y espionaje a través de redes sociales. También son populares los ataques DoS/DDoS, que pueden ser usados para interrumpir los servicios. A menudo algunos usuarios autorizados pueden también estar directamente involucrados en el robo de datos o en su mal uso. Pero si se toman las medidas adecuadas, la gran mayoría de este tipo de ataques pueden prevenirse, por ejemplo, a través de la creación de diferentes niveles de acceso, o incluso limitando el acceso físico. Las medidas de **seguridad informática** que se pueden tomar incluyen:

1. Asegurar la instalación de software legalmente adquirido: por lo general el software legal está libre de troyanos o virus.
2. Actualizar el software y el sistema operativo: esto significa que aprovechará las últimas revisiones de seguridad.
3. Utilizar software antivirus: las soluciones de seguridad detectarán y eliminarán las amenazas. Mantenga su software actualizado para obtener el mejor nivel de protección, con las reglas de configuración y del sistema adecuadamente definidos.
4. Hardware y software cortafuegos: los firewalls ayudan con el bloqueo de usuarios no autorizados que intentan acceder a tu computadora o tu red.
5. Uso de contraseñas complejas y grandes: las contraseñas deben constar de varios caracteres especiales, números y letras. Esto ayuda en gran medida a que un hacker no pueda romperla fácilmente.
6. Cuidado con la ingeniería social: a través de las redes sociales los ciberdelincuentes pueden intentar obtener datos e información que pueden utilizar para realizar ataques.
7. No abrir archivos adjuntos de correos electrónicos de remitentes desconocidos: podrían estar infectados con malware. No hacer clic en los vínculos de los correos electrónicos de remitentes o sitios web desconocidos.
8. Criptografía, especialmente la encriptación: juega un papel importante en mantener nuestra información sensible, segura y secreta.
9. Evitar el uso de redes Wi-Fi no seguras en lugares públicos: las redes no seguras lo dejan vulnerable a ataques del tipo “Man-in-the-middle”.
10. Tapar la webcam. En general, los distintos sensores de un dispositivo como GPS, micrófono, acelerómetro, etc. podrían ser hackeados y utilizados para fines ilícitos.

Referencias: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/brocker

6.3 VIRUS

6.3.1 Virus – dificultades que generan

¿Si un virus no hace más que copiarse a sí mismo y nada más, por qué trabajan las personas tanto cuando tienen uno?. Por ejemplo: un virus emite pitidos cada vez usted le da a una tecla con su teclado, el 18 de cada mes, ¿por qué hace que todos quienes nos infectemos queramos librarnos de él?. Hay varias razones para esto. Si usted no se libra del virus, hay una probabilidad fuerte que en el futuro usted se lo pasará a un proveedor o cliente que serán perturbados. Más de 99% de los virus que realmente se extendieron es residente en memoria. Hay la posibilidad de incompatibilidades entre el virus y algún otro programa que usted está corriendo [por ejemplo, Jerusalem intenta usar uno de los recursos que Novell NetWare usa, así que usted no puede ejecutar Novell NetWare y Jerusalem en el mismo sistema]. Además, aunque nosotros hemos analizado y hemos documentado el virus, el usuario podría permanecer nervioso pensando que el virus hace más de lo que se ha documentado, o que él tiene una versión diferente del virus. Así en casi todos los casos que nosotros hemos visto, las personas realmente quieren librarse del virus [a excepción, a veces, por guardar un espécimen para un “zoológico”].

Si usted desea realmente librarse del virus, esto le va a tomar tiempo, y tiempo es dinero. Cada PC tomará varios minutos por lo menos, y cada disco flexible tomará varios segundos por lo menos [y puede haber muchos discos flexibles]. Sería bueno si usted sólo tuviera que preocuparse por esas computadoras y discos flexibles que se infectan, pero por supuesto usted no sabe cuáles son éstos hasta que haya hecho la búsqueda de virus, así que tiene que verificar todo.

6.3.2 Cosas que no son virus

ERRORES DE PROGRAMA. (BUGS)

Los Bugs no son virus, y los virus no son "Bugs". Yo estoy usando un procesador de textos que sé que tiene un "Bug" serio. Si usted trabaja durante mucho tiempo con un archivo grande, entonces en el futuro algo sale mal dentro del programa, y se niega a permitirle guardar el archivo al disco, de manera que usted perderá todo el trabajo que ha hecho desde la última vez que salvó. Este programa viene de una casa de software importante, y ellos no hicieron esto a propósito, así que no es un troyano. Los programadores cometieron un error. Los programadores son humanos, y los humanos cometen errores. Los programadores, como otros humanos, tienen orgullo, y no les gusta admitir que cometen errores, así que ellos los llaman "bugs", como si el bicho (bug) hubiese entrado volando a través de la ventana y se hubiese establecido en el programa. Todos los programas suficientemente complejos tienen "bugs".

El software anti-virus no descubre "bugs". Si lo hiciera, informaría "bugs" en casi todo. El término inglés bug, que significa insecto o, más genéricamente bicho, tiene su origen en los primeros días de la informática, cuando se descubrió que un problema de hardware en un ordenador electromecánico de la Universidad de Harvard (EEUU) se debía a una polilla atrapada entre los contactos de un relé de la máquina.

ALARMAS FALSAS

Las alarmas falsas no son virus. Una alarma falsa es cuando usted piensa que tiene un virus, pero está equivocado. A veces, las personas tienen alguna falla del hardware o del software, y después de ejecutar algunos diagnósticos, eliminada la posibilidad de hardware o problemas del software, concluye que es por consiguiente un virus, y procede en esa asunción. Más a menudo, una alarma falsa es el resultado de la ejecución de software anti-virus.

El software anti-virus, igual que otro software, no es infalible. Los dos errores principales que un programa anti-virus puede cometer son: no encontrar un virus que está allí [yo probé una vez un programa que no encontró ningún virus de sector de arranque, y éstos se consideran causa de más de dos tercios de las infecciones] o indicar que un virus está presente cuando no hay ninguno, y eso se llama una alarma falsa.

Cuando un programa anti-virus da una alarma falsa, se parece bastante mucho a la realidad.

Hay varias cosas que podrían indicar que la alarma es falsa:

- Sólo un archivo está dando la alarma [o quizás cuatro archivos, pero ellos son copias del mismo archivo].
- Sólo un producto da la alarma; otros productos dicen que el sistema está limpio.
- Usted recibe la alarma después de ejecutar productos múltiples, pero no al arrancar en frío y ejecutar un producto.
- El virus que se descubre está en la lista de virus reconocidos [por supuesto, esta lista cambia todo el tiempo].

No hay ninguna regla fija que puede aplicarse, desgraciadamente. Usted no puede decir que es una alarma falsa si uno de los cuatro casos indicados arriba es verdad, o incluso si los cuatro son verdad. La única manera de realmente descartar una alarma falsa es enviarle el archivo sospechoso al vendedor del producto que da la alarma, pidiéndole que verifique que es un virus, analizándolo en su laboratorio de virus. Y esta determinación tarda algún tiempo.

Entretanto, una alarma falsa puede ser tanta molestia como un virus real, o más aún. Si tiene un disco flexible que se infecta con el virus Stoned, usted puede copiar los datos simplemente fuera del disco y puede destruir el disco flexible infectado, o usted puede librarse del virus con su programa de anti-virus, o exigir un disco de reemplazo. En cualquier caso, el costo es sólo unos segundos. Pero si usted consigue una actualización de su programa de anti-virus favorito, y le dice que usted tiene el virus Stoned en los archivos en su servidor de archivos, entonces resolver la situación tomará mucho más tiempo. Usted sabe y yo sé que Stoned no puede infectar archivos. Pero simplemente quizá alguien ha escrito un virus de archivo y ha decidido llamarlo Stoned [por ejemplo, hay dos virus no relacionados, uno llamado Parity y el otro Parity.b]. Así que, usted le envía el archivo a su vendedor del producto para el análisis y comentario. Entretanto, para estar seguro, usted quita el archivo del servidor [o posiblemente el producto está obstruyendo el acceso a ese archivo]. Esta situación significa que algún sistema importante no trabajará más, cuando necesite ese archivo.

No es sorprendente que algunas personas hayan cambiado su software anti-virus después de demasiadas alarmas falsas.

El software anti-virus no descubre alarmas falsas. ¿Si lo hiciera, no informaría la alarma falsa?

CHISTES (Jokes)

Un chiste es algo cómico. Por supuesto, que lo que una persona encuentra cómico no es igual que lo que otra persona encuentra cómico. Depende de su sentido del humor. Considere un programa que pretende formatear su disco duro, y luego revela que no tiene formato. ¿Es eso cómico?. Depende de su sentido de humor.

Algunas personas aman hacer bromas, y en ciertas fechas uno debe aplicar un poco de escepticismo a los informes de virus. Algún software anti-virus descubre chistes, y le dice "Usted tiene un chiste llamado. . . ". La razón para esto es que algunos chistes están bastante extendidos,

y se conciben para causar preocupación, así que el programa anti-virus está intentando tranquilizar las cosas, diciendo "Si, yo sé sobre esto, y es inofensivo".

TROYANOS

Un troyano es un programa que hace algo más de lo que el usuario estaba esperando; y esa función adicional es perjudicial. Esto conlleva a un problema en la detección de troyanos. Suponga que yo escribí un programa que podría descubrir infaliblemente si otro programa formateó el disco duro. ¿Entonces, puede decir que este programa es un troyano? Obviamente no si se supusiera que el otro programa formateaba el disco duro [como el FORMAT hace, por ejemplo], entonces no es un troyano. Pero si el usuario no estuviera esperando el formateo, entonces es un troyano. El problema es comparar lo que el programa hace con las expectativas del usuario. Usted no puede determinar las expectativas del usuario para un programa.

Es así que nosotros tenemos que hacer algunas consideraciones. Generalmente se considera que el Diskette de Información del SIDA es un troyano. Aproximadamente 20.000 copias de éste se mandaron por correo a los usuarios en 1989, pretendiendo ser un programa que le enseña sobre el virus del SIDA. De hecho, era un troyano; después de que usted arrancaba su computadora 90 veces, él encriptaba y escondía todos los nombres de archivo de su disco duro, y demandaba que usted pagase por su licencia para usarlo. Aunque la documentación que vino con él le decía que era probable que algo malo pasara, generalmente se considera que es un troyano. El FORMAT no es un troyano.

Como regla, usted no ve muy a menudo troyanos. Ellos no se copian, y no se extienden de la manera que los virus hacen. Los troyanos no son una amenaza real, excepto en una de las circunstancias siguientes.

- Cuando ellos se diseminan ampliamente, como el Diskette de Información de SIDA.
- Ellos son focalizados en una organización, en este caso es un "trabajo interno", hecho por un empleado.

Algunos productos anti-virus descubren unos troyanos [como SIDA], pero la mayoría de los productos no descubren troyanos en absoluto.

PROGRAMAS ADULTERADOS

Algunos archivos simplemente se adulteran [quizás por un problema del hardware], y cuelgan a la computadora cuando corren. Por alguna razón, éstos a veces terminan en colecciones de virus, a menos que la colección sea mantenida cuidadosamente.

INTENCION DE HACER VIRUS

Algunos autores de virus son menos habilidosos de lo que les gustaría ser, y escriben lo que se piensa claramente que es un virus, pero por alguna razón hay semejante error de programación que el virus no trabaja en absoluto. Ellos sueltan éstos, sin embargo, en la creencia aficionada que nadie los probará [o quizás ellos no los probaron]. Un error típico es confundir el decimal con el hexadecimal, y así que su código fuente dice probablemente `int 21' para la función interrupción del DOS, pero debería haber dicho `int 21h ' [que es 33 en decimal].

CUENTAGOTAS (DROPPERS)

Un cuentagotas es un programa que no es un virus, ni se infecta con un virus, pero cuando corre instala un virus en la memoria, en al disco, o en un archivo. A veces se han escrito cuentagotas como un portador conveniente para un virus, y a veces como un acto de sabotaje. Algunos programas anti-virus intentan descubrir cuentagotas.

GÉRMENES

Un germen es un caso de virus de generación cero, y en semejante forma que la infección no pudiera pasar naturalmente. Por ejemplo, un virus que sólo infecta archivos más grandes que 5Kb, pero ha infectado un archivo diminuto de 10 b. Alternativamente, podría ser un caso de virus sin ningún archivo anfitrión. Si usted quita el virus, usted queda con un archivo de cero-byte. Éste es el archivo original creado por el autor del virus.

GUSANOS

Los Gusanos (Worms) son programas que realizan copias de sí mismos, alojándolas en diferentes ubicaciones del ordenador. El objetivo de este malware suele ser colapsar las computadoras y las redes informáticas, impidiendo así el trabajo a los usuarios. A diferencia de los virus, los gusanos no infectan archivos.

6.3.3 Tipos diferentes de virus

VIRUS DEL SECTOR DE ARRANQUE (BSVs)

Los tipos mas comunes de virus son virus de sector de arranque [BSVs], como Form o Stoned. Éstos infectan los sectores de arranque de discos flexibles, y/o el sector de la partición [Registro Maestro de Arranque, MBR] o el sector de arranque del DOS [Registro de Arranque del DOS, DBR] de discos duros. He aquí como un BSV se disemina.

Un disco flexible recién ha llegado, con algunos datos en él [algunos archivos de procesador de textos y una hoja de cálculo, quizás]. Ésta es parte de un proyecto que usted está haciendo juntamente con un colega. Lo que su colega no sabe es que su computadora está infectada con un BSV, y por consiguiente lo está también el disco que él le envió. Usted puso el disco en la unidad A y comenzó a usar estos archivos. Hasta ahora, el virus no ha hecho nada. Pero cuando usted termina por ese día, usted apaga a la computadora y va a casa. El próximo día, usted entra y enciende. El disco flexible todavía está en la unidad A, así que la computadora intenta arrancar de este disco. Carga el primer sector en la memoria y lo ejecuta [normalmente, esto es un pequeño programa escrito por Microsoft para cargar el DOS; o si no puede encontrar DOS en el disco, para decirle así: No hay disco de sistema o error de disco. Reemplácelo y oprima una tecla cuando esté listo (Non-System disk, or disk error. Replace and press any key when ready)]. Todos hemos visto este mensaje numerosas veces, así que usted abre la puerta de la unidad A y aprieta una tecla.

Pero este disco está infectado con Stoned, así que lo que ejecutó no era exactamente el programa de Microsoft, sino el virus Stoned, escrito en 1987 en Nueva Zelanda [y a veces llamado el virus de Nueva Zelanda]. El virus se instala en el disco duro, reemplazando el MBR, y copiando el MBR original un poco más adelante en el disco.

Cuando usted pone en marcha el disco duro, el MBR corre, pero esto es el virus Stoned. El virus Stoned queda residente de memoria, capturando la interrupción 13h, de lectura/escritura del disco, y entonces carga el MBR original, y el proceso arranque continúa normalmente. Pero, como la interrupción de read/write de disco fue capturada, cada vez que haya un acceso de escritura o lectura [usted piensa que está haciendo una lectura, pero el virus, en cambio, decide escribir] a la unidad A, el disco flexible se examina, y si todavía no está infectado, el virus Stoned se instala en el sector de arranque. Así, su computadora está infectando cada disco puesto en la unidad A, y más pronto o más tarde de éstos se enviarán a un colega, y el ciclo continuará.

El detalle de varios BSVs es diferente, pero el principio es el mismo. Ellos son llevados por los sectores de arranque de discos infectados, y sólo de esa manera [un BSV no puede

extender por una red, por ejemplo]. Y la única manera a infectarse es intentar arrancar de un disco infectado, aún cuando el arranque falle.

Los BSVs infectan PCs. Ellos no tienen en cuenta que sistema operativo está corriendo, o qué software de seguridad se instala, porque en el momento en que el BSV se instala el sistema operativo o el programa de seguridad no está corriendo todavía. Sin embargo, con algunos sistemas operativos no-DOS por ejemplo, Windows NT, u OS/2], aunque el PC es infectado, el virus no puede copiarse a si mismo hacia otros discos y no se puede difundir. Puede, sin embargo, todavía hacer daño, como fue descubierto a través de un sorprendido usuario de UNIX cuando Michelángelo se activó el 6 de Marzo.

A la mayoría de las personas, el hecho que los virus pueden infectar de esta manera les sorprende, lo cual da cuenta porqué los BSVs son tan comunes.

VIRUS DE MACRO

Los virus de Macro [como el WM.Concept], el último desarrollo de virus, parece volverse una amenaza significante, por varias razones.

Macros, escritos en WordBasic, y accesible a muchos usuarios de computadora, son más fáciles de escribir que los "tradicionales" virus de archivo [escritos, en su mayoría, en código assembly].

Ellos fueron los primeros virus en infectar archivos de datos, en lugar de ejecutables. Los archivos de datos, a los cuales se atan macros, proporcionan un método de replicación más eficaz a los virus que los archivos ejecutables. Frecuentemente se intercambian archivos de datos mucho más que archivos ejecutables. Si usted agrega el incremento del uso de e-mail [y la habilidad de atar archivos al e-mail], y el acceso masivo a Internet [y servicios en-línea como CompuServe y América Online], es probable que esto haga a los virus de macro una amenaza mucho mayor para los usuarios de computadora que los virus de archivo tradicionales.

Los virus de macro no son de una plataforma específica. Hay versiones de Microsoft Word para Windows 3.x, Windows 95, Windows NT y Macintosh. Esto hace a todos estos sistemas operativos susceptible a los virus de macro [aunque algo en un macro que produce llamadas a un sistema operativo específico [como con el WM.FormatC macro trojan] se restringirá a ese sistema operativo en particular].

Los virus de macro ya han tenido un marcado efecto. WM.Concept actualmente contabiliza alrededor de 50% de todos los informes de virus a nuestra sección de asistencia técnica. Y aunque WM.Concept no causa daño a los datos, nosotros ya hemos visto el primero de los pasos [aunque vacilante] hacia un virus de macro que amenaza datos.

Los virus de macro no están restringidos a Word de Microsoft para Windows. En Enero de 1996, aparecía el primer virus de macro para infectar archivos Lotus AmiPro [APM.GreenStripe]. Al contrario de lo que ocurre en Word para Windows en el que se unen macros directamente a DOC [y DOT] los archivos, los macros de AmiPro están contenidos en un archivo separado [con la extensión SMM]; esto hace posible intercambiar documentos AmiPro [por ejemplo, vía e-mail] sin intercambiar macros infectados. Y XM.Laroux que aparecía en Julio de 1996, es el primer virus de macro para infectar las hojas de cálculo Microsoft Excel para de Windows.

VIRUS DE ARCHIVO TSR

Los virus de archivos TSR son ahora poco comunes. Como el nombre lo sugiere, éstos infectan archivos. Éstos normalmente son COM y EXE, pero hay algunos virus de drivers de

dispositivos, y algunos virus infectan archivos overlay (OVL); los programas ejecutables no siempre tienen la extensión COM o EXE, aunque así sea en más de 99% de ellos.

Para que un virus de TSR se disemine, alguien tiene que ejecutar un programa infectado. El virus queda residente en memoria, y típicamente mira cada ejecución del programa después de lo cual lo infecta, si es que ya no lo estaba. Algunos virus se llaman "fast infectors" (que infectan inmediatamente) y ellos infectan apenas usted abre el archivo [por ejemplo, un backup podría abrir cada archivo en el drive]. El Dark Avenger fue el primer "fast infector". En el caso de Green Caterpillar, el disparador de infección es algo que determina que archivos están presentes [como DIR]. Se han usado otros disparadores, pero el más común es infectar cada programa que usted está a punto de correr.

VIRUS DE ARCHIVO NO-TSR

Es mucho mas fácil escribir un virus NO-TSR, y así lo hacen tantos de los autores de virus principiantes. Pero es bastante raro encontrar semejante virus en la actualidad; menos del 1% de eventos informados es un virus NO-TSR. Con este tipo de virus, ejecutando un programa infectado se ejecuta el virus, que busca otro archivo para infectar, y lo infecta. Vienna es el mas común de los virus NO-TSR. Vienna fue el primer virus de archivo, pero ahora tiene el estado de "raro".

Hay muchos virus basados en Viena, porque en 1987 se publicó en un libro un desensamblado [que es casi equivalente al código de la fuente].

VIRUS COMPAÑEROS

Si usted tiene un archivo de COM y un archivo de EXE con el mismo nombre de archivo y teclea ese nombre, DOS corre los archivos COM preferentemente a los archivos EXE. Los virus compañeros usan este rasgo de DOS. Cada archivo de EXE que usted tiene adquiere a un compañero COM con el mismo nombre. Entonces, cuando usted intenta ejecutar su programa EXE, realmente corren los programas COM, y ése es el virus. Cuando el virus ha terminado haciendo lo que quiere hacer [como crear a otro compañero para otro archivo], corre entonces los programas EXE, para que todo parezca trabajar normalmente.

Ha habido unos virus compañeros exitosos, pero no muchos. La ventaja principal para el autor del virus es que debido a que el archivo EXE no cambia, algún software que busque cambios no podría detectar que un virus está extendiendo.

Otro tipo de compañero es el compañero de PATH (camino, ruta de directorios). Esta clase de virus pone un programa en un directorio que está antes en el PATH de DOS del que está la víctima. Cuando usted ejecuta un programa que no está en un sub-directorio actual, DOS busca el programa en varios sub-directorios, tal como lo especifica el orden de la directiva PATH en el archivo de AUTOEXEC.BAT. Los compañeros de PATH son más difíciles de escribir que los compañeros ordinarios y no hay muchos de ellos.

VIRUS DE SOBRESCRITURA

Un virus de sobrescritura borra cada archivo que infecta, reemplazándolo con sí mismo simplemente, y el programa ya no funciona. Debido a que esto es tan obvio, estos virus nunca han sido exitosos para distribuirse.

VIRUS MULTIPLES

Algunos virus, como Tequila, infectan objetos múltiples. Cuando usted ejecuta un EXE infectado por Tequila, Tequila se instala en el MBR. Cuando usted arranca la computadora, Tequila se ejecuta en el MBR, y va residente a memoria. Mientras Tequila está residente en la

memoria, infecta archivos EXE. Otros virus, como algunas de las versiones de Anticad, infectan COM, EXE y MBRs indistintamente. Algunos virus infectan COM, EXE, MBRs y drivers de dispositivo.

OBJETOS MISCELANEOS DE INFECCION

Hay un virus que infecta archivos OBJ. Hay un virus [Starship] que infecta creando un nuevo DBR, dejando el viejo intacto, dejando el código en el MBR intacto, y cambiando el puntero en el MBR para que el DBR de Starship se ejecute ante el DBR original.

Hay otros virus [DIR II y Dir.Byway] que infectan archivos del sistema cambiando la FAT y directorios para que los archivos en el disco duro queden con enlaces cruzados al virus.

6.3.4 Características de los virus

RAPIDO

Como explicó anteriormente, un "fast infector" se extiende rápidamente dentro de una computadora infectando todo lo que se accede. Un infector rápido no es tan malo como parece; es tan fácil limpiar una computadora con 1.000 archivos infectados como una con 10, con tal de que usted tenga un programa de anti-virus que haga un trabajo de limpieza bueno. Sin embargo, la mayoría los productos anti-virus verifican la memoria buscando virus, y la posibilidad de que un infector rápido esté en memoria es una de las razones de hacerlo. Si hay un infector rápido en memoria y el producto abre todos sus archivos, usted infectaría cada archivo en la computadora infectada.

LENTO

El opuesto de un "fast infector" es un "show infector" (infector lento). La idea aquí es que si el virus se extiende despacio, usted tiene menos probabilidades de notarlo y matarlo. Hay varias maneras que un infector lento pueda trabajar, pero el infector lento clásico trabaja sólo infectando esos archivos que usted ya había pensado cambiar. Esto significa que si usted está ejecutando un detector de cambios como una medida anti-virus, el detector de cambios avisará cada vez que hay una infección, pero como usted había pensado modificar el archivo, le dirá que acepte el cambio.

Starship es otra manera de hacer un infector lento. Sólo infecta archivos cuando usted los copia a su disco duro. Así que ningún archivo en el disco duro cambia, y el detector de cambios está contento. Pero cuando usted copia un archivo a un disco flexible, la copia se infecta, y cuando usted lleva esto a otra computadora protegida por el detector de cambios, el detector de cambios lo advierte de la existencia del nuevo archivo. Usted tranquilizará al detector de cambios indicándole que usted sabe que es un nuevo archivo, el descubridor de cambio estará contento, y otro sistema se habrá infectado.

OCULTO (STEALTH)

Si un virus es residente en memoria [como es más de 99% de los virus], entonces él ha enganchado una de las interrupciones, por lo menos. Si es un BSV, entonces ha enganchado la interrupción 13h, de lectura/escritura en disco. Si es un virus oculto y cualquier programa intenta leer el sector de arranque, entonces el virus dice "Aha, alguien quiere ver el sector de arranque; yo leeré entonces el sector de arranque original de donde yo lo puse, y presentaré ese en su lugar". Así que el software no ve nada excepcional. Brain, cosecha 1986, fue el primer virus que usó este truco. Los virus de archivo pueden usar un truco similar para enmascarar su presencia, para que cualquier software que sólo lee el archivo vea que los bytes que estaban allí antes de

que el virus llegara. Frodo es un ejemplo de esto. Es mas común ver virus ocultos en BSVs que en virus de archivo, ya que mas fácil para un autor de virus realizar ocultamiento en un BSV.

POLIMORFISMO

El tipo mas común de programa anti-virus que usan las personas en el escáner, buscando por un repertorio de virus. Así para la mayoría de autores de virus éste es el tipo de producto que mas le gusta derrotar. Un virus polimórfico tal que si usted toma dos instancias del virus, no hay ningún byte en común entre ellas, así que usted no puede anotar una sucesión de bytes y luego buscarlos en ese orden para descubrir el virus. Usted tiene que hacer algo mucho más complejo y difícil.

6.3.5 Daño hecho por los virus

Nosotros podemos categorizar el daño hecho por virus en seis grupos, según la severidad del daño. Algunas autoridades en la materia postulan la posibilidad de un virus que realmente hace algo bueno, pero nadie ha demostrado semejante virus todavía.

Nosotros definimos daño como: el virus hace algo que yo lo preferiría que no haya hecho. Y nosotros cuantificamos daño midiendo cuánto tiempo toma para volver a poner las cosas de la manera que debe ser.

No incluimos el daño "consecuente" en esta categorización [daño hecho por el usuario en un esfuerzo equivocado por librarse del virus]. Es notable cuantas personas darán formato al disco duro para librarse de Stoned, por ejemplo. Todo esto sólo ha conseguido que pierda todos sus datos. El virus está intacto, ya que reside en el MBR que no está afectado por FORMAT. Ni podemos incluir el daño hecho por incompatibilidades oscuras entre el virus y el sistema. Por ejemplo, si una computadora que estaba originalmente preparada para DOS 2 [pero está ejecutando una versión posterior de DOS ahora] se infecta por Stoned, entonces un número grande de archivos se adulterará porque el diseño del virus no había anticipado esta situación.

DAÑO TRIVIAL

Esto es hecho por un virus como Form [una vez fue el virus mas común en el mundo]. En el 18 de todos los meses, cada tecla que usted oprima hace al parlante emitir una señal sonora. Todos que usted necesita es librarse del virus. Esto normalmente tomará segundos o minutos [por cada computadora].

DAÑO MENOR

Un buen ejemplo de daño menor es la manera que el virus de Jerusalén borra cualquier programa que usted intenta correr después de que el virus ha ido de residente en memoria, los viernes 13. En el peor de los casos, usted tendrá que re-instalar algunos programas, así que recuperar el daño demandará unos 30 minutos por computadora.

DAÑO MODERADO

Si un virus da formato al disco duro, mezcla la FAT o borra el disco duro, éste es daño moderado. El daño es sólo moderado porque usted sabe que ha pasado, y usted puede re-instalar DOS y recargar el backup de ayer, porque usted hace un backup todos los días. Así que, usted perderá en mitad del promedio el trabajo de un día, más quizá una hora que para reinstalar y restaurar. El virus más famoso de daño moderado es Michelángelo.

DAÑO MAYOR

Este ocurre cuando un virus ataca tanto sus backups como su disco duro. Cada 16ava vez que un archivo infectado con Dark Avenger se ejecuta, sobrescribe un sector al azar en el disco duro con "Eddie lives . . . somewhere in time". Esto podría haber estado ocurriendo durante varias semanas. Usted descubre al Dark Avenger, se libra del virus, y halla "Eddie lives . . . somewhere in time" en varios archivos. Usted restaura el backup de ayer, y halla "Eddie lives . . . somewhere in time" en aquéllos también. Usted podría tener que remontarse unas semanas atrás, antes de que pueda encontrar archivos de datos limpios, y cuando usted ha restaurado con un backup de seis semanas de antigüedad, encontrará que no tiene ninguna manera de rehacer el trabajo, porque no tiene los documentos originales con los que trabajó.

DAÑO SEVERO

Daño severo ocurre cuando un virus hace cambios graduales y progresivos [para que también se adulteren backups], pero los cambios no son obvios [no hay "Eddie lives . . . somewhere in time" para buscar]. Usted termina simplemente no sabiendo si sus datos son correctos o cambiaron.

DAÑO ILIMITADO

Algunos virus [como Cheeba, Vacsina.44.login y GP1] apuntan a conseguir la contraseña del administrador del sistema y pasarlo a una tercera persona. Por ejemplo, en el caso de Cheeba, crea a un nuevo usuario con máximo privilegio, con un nombre de usuario fijo y contraseña. El daño es hecho entonces por esa tercera persona, que puede ingresar (log-in) al sistema y hacer lo que desee.

6.3.6 Como se difunden los virus

Parece ser una creencia común que los virus son difundidos por juegos, por shareware o por BBS (Bulletin Board System). La verdad es más compleja. Primero, recuerde como trabaja la clase más común de virus, los virus de sector de arranque. Un disco flexible físico tiene que estar involucrado y no necesita que haya software en él. Usted no puede conseguir un BSV usando un BBS.

Las rutas más probables por las que un virus entra en una organización son los ingenieros y los padres.

Los ingenieros de hardware visitan un número grande de computadoras, y como la pequeña abeja ocupada, podría recoger un poco de polen aquí, y depositarlo allí. Los ingenieros de hardware deberían tener todos sus discos de software permanentemente protegidos contra escritura, pero no lo hacen. Los ingenieros de hardware frecuentemente deberían verificar cualquier disco con escritura habilitada para ver si tiene virus, pero no lo hacen. Por supuesto, la mayoría de los ingenieros de hardware está limpia y bien educada, pero hay algunos que requieren reeducación.

Los padres tienen niños, y si hay un PC en casa, y los niños son adolescentes jóvenes, entonces ellos posiblemente intercambien software en la escuela. Los discos que ellos traen casa bien podrían estar infectados, y si el padre está llevando y trayendo discos del trabajo, ellos podrían tomar un virus fácilmente y llevarlo al trabajo con ellos.

Un virus de sector de arranque podría llegar en un disco de datos de un colega.

Otras maneras de conseguir un virus incluyen:

- en software empaquetado [algunas de las compañías más grandes han enviado un virus accidentalmente en software empaquetado];
- junto con el hardware comprado [la mayoría el hardware viene con discos que contienen utilidades o drivers];
- los vendedores que ejecutan demostraciones podrían instalar el virus que ellos recogieron inconscientemente del último lugar donde ejecutaron su demostración.

6.4 MEDIDAS DE PREVENCION

Nosotros recomendamos que todo sea verificado antes de que se use. Esto incluye discos, memorias SD, pen drives, etc. Esto podría hacerse con un escáner como AVG, el cual podría instalarse en cada computadora [si es conveniente, porque si no lo es, no se hará] o podría instalarse en determinadas computadoras dedicadas, lo que es más conveniente para las personas de soporte, para mantenerlo actualizado, pero menos conveniente para los usuarios.

Alternativamente, usted puede hacer todo esto tan transparente y sin dificultad instalando un escáner residente. Windows incorpora el antivirus Windows defender por defecto y también un firewall. Esto significa que todo se examina automáticamente sin que el usuario sea consciente de ello [a menos que sea encontrado un virus, por supuesto]. El escáner residente es la ruta que la mayoría de las personas escogen, junto con algunas máquinas dedicadas.

6.4.1 Reglas, procedimientos, educación y herramientas

Linda había trabajado en su trabajo durante algunos años. Ella tomó trabajo de vez en cuando para hacer en la PC de su casa, con el conocimiento y aprobación de su supervisor. Un día, la sección de soporte de PC encontró un virus en la PC de su oficina. El mismo día, un poco mas tarde, Linda fue despedida por introducir un virus a la empresa.

Linda estaba muy disgustada por lo que ella vio como despido injusto y demandó a la compañía. Ella ganó, porque la compañía para la que ella trabajó no tenía ninguna regla para decirles a los empleados qué hacer [así que ella no había roto ninguna regla]. Aunque ellos tenían software anti-virus, no había ningún procedimiento para verificar los discos que ingresaban [es así que ella no faltó a procedimientos de la compañía].

En la investigación, parecía probable que la sección de soporte de PC había infectado su máquina accidentalmente, y sólo lo descubrió cuando enviaron discos, copiados en la máquina de Linda, a otra compañía que verificó la presencia de virus y encontró uno en esos discos.

Si tales procedimientos hubieran existido y Linda los hubiese ignorado, entonces la compañía habría tenido una buena razón para despedirla. Por supuesto, con reglas apropiadas y procedimientos, probablemente no habrían sido infectados por un virus desde el primer momento.

Pero usted tiene que reconocer el hecho de que las personas se comportan la manera que lo hacen. Si usted hace sus procedimientos anti-virus onerosos y difíciles, probablemente se ignorarán en los sitios donde los virus son muy raros, y el costo y molestia de los procedimientos es demasiado grande.

Un buen conjunto de reglas podría ser como sigue:

- Cualquier unidad extraíble entrante debe verificarse.
- Si su software anti-virus encuentra un virus, avise al soporte de PC.

Nótese que las reglas son muy simples. De esa manera es más probable que, las personas las recuerden y respeten. La próxima cosa que usted necesita son procedimientos. El procedimiento les dice a los usuarios cómo obedecer las reglas. El procedimiento para verificar discos debe redactarse con detalles [' Ponga el disco/unidad USB en el drive, y tipee. . . ']. Si usted tiene una computadora que actúa como servidor, ponga el procedimiento en la pared, cerca de ella.

La educación también es importante. Usted no puede decirles a adultos que hagan algo y esperar simplemente que ellos le obedezcan sin preguntar; tiene que explicarles las razones. Puede hacer esto con charlas, o consiguiendo el vídeo sobre virus de Dr Solomon y permitiéndoles mirarlo.

Usted también tiene que proporcionar herramientas. No puede descubrir un virus con sus manos desnudas. Cualquier estrategia de anti-virus sensata debe tomar cuenta del hecho que incluso un usuario capacitado es falible; y que ellos transgredirán incluso las mejores reglas y procedimientos [consciente o inconscientemente. . . recuerde que la seguridad no es la preocupación primaria de personal que trabajo en Ventas, Comercialización y otras secciones dentro de una organización]. La base de cualquier estrategia de anti-virus coherente, por consiguiente, deben ser herramientas de anti-virus que eficazmente deberán descubrir, quitar y prevenir infecciones de virus. . . incluso cuando no se han seguido las reglas y procedimientos.

6.4.2 Herramientas anti-virus

ESCÁNERES

Un escáner es un programa que sabe encontrar un repertorio particular de virus. Los escáneres se actualizan en forma mensual o trimestral.

Los escáneres pueden ejecutarse a pedido (on-demand) o en cada acceso a los archivos (on-access). FindVirus es un escáner que se ejecuta a pedido, y debe ser corrido por el usuario [aunque esto podría hacerse automáticamente, al arrancar, indicándolo en el AUTOEXEC.BAT; o usando un programador de tareas (scheduler)].

WinGuard [Windows 3.x, Windows 95 y Windows NT] y AVG [Windows 7 hasta 10] son escáneres que trabajan continuamente, cuando se accede a los archivos. Cada vez que algún disco se accede, se verifica que no tenga virus el sector de arranque; y cuando se usa cualquier archivo, se verifica que el mismo no tenga virus.

Ambos programas pueden ser [opcionalmente] configurados para verificar archivos cuando se escriben al disco duro [útil si están transmitiéndose archivos de un sitio remoto, como un BBS, o Internet]. VirusGuard ocupa aproximadamente 9 kB de memoria convencional [DOS]; WinGuard, que es un programa específico para Windows no usa memoria convencional. Cualquier tiempo adicional involucrado para verificar el disco o el archivo es improbable que sea percibido en la mayoría de los casos. VirusGuard, un programa residente DOS, no tiene toda la capacidad de FindVirus [para economizar en consumo de memoria y tiempo]; específicamente, VirusGuard no puede encontrar virus de macro [que no trabajan, sin embargo, bajo DOS] y un porcentaje sumamente pequeño de los virus polimórficos [VirusGuard encontrará virus polimórficos en memoria, si se ejecutó un programa infectado]. WinGuard que no tiene los condicionamientos de un programa residente DOS, tiene la misma capacidad de detección que FindVirus.

Nota: Si en su PC está instalado un antivirus que monitorea todos los ejecutables, en caso de detectarse un archivo que contenga virus aparecerá una pantalla similar a la anterior.

CHECKSUMMERS

Un checksummer es un descubridor de cambios. Los archivos ejecutables no deben cambiar, salvo por una buena razón, como una actualización de software. Un checksummer intenta descubrir cambios. La ventaja de los checksummers es que no descubren un repertorio de virus, así que no necesitan actualización. El aspecto negativo de los checksummers, es que son más molestia que los escáneres [los archivos en su computadora cambian más a menudo de lo que usted podría haber pensado, por razones buenas y válidas], y no descubren todos los virus. Por ejemplo, los checksummers no descubren los infectores lentos; no descubren todos los virus de sector de arranque [si el código del disco duro queda inalterado]; y tienen problemas con virus ocultos. Algunas personas usan checksummers, pero son una minoría.

Los checksummers pueden ejecutarse a pedido [como ViVerify], o en cada acceso a los archivos.

SOLUCIÓN COMPLETA DE SEGURIDAD

AVG Internet Security

- Antivirus avanzado: Analiza su PC en busca de virus, ransomware, spyware y otros tipos de malware.
- Escudo de comportamiento: Envía alertas si detecta un comportamiento sospechoso del software de su PC.
- Detección de IA: Identifica de forma proactiva muestras de malware para protegerle contras las amenazas nuevas.
- CyberCapture: Bloquea las amenazas nuevas haciendo que nuestro software antivirus las envíe para su análisis.
- Análisis de PUA: Identifica las aplicaciones potencialmente no deseadas que pueda haber descargado de forma inadvertida.
- Análisis turbo: Reduce el tiempo de análisis omitiendo los archivos que ya sabe que son seguros
- Actualizaciones en tiempo real: Asegura que siempre esté protegido, con actualizaciones automáticas de seguridad, resolución de errores y nuevas funciones.

- Modo de «no molestar»: Detiene automáticamente todas las ventanas emergentes y las notificaciones de la aplicación, para que pueda concentrarse en lo que está haciendo.
- Modo silencioso: Pospone los análisis, las actualizaciones y las ventanas emergentes para evitar las interrupciones mientras trabaja.

6.4.3 Redes y virus

Una red es un grupo de computadoras conectadas para hacer más fácil compartir datos. Esto da oportunidades interesantes a los virus, y para tratar con ellos.

Hay una percepción común que una vez que un virus ingresa a una red, de algún modo infecta muy rápidamente la red entera. La verdad, por supuesto, es más compleja. Primeramente, los BSVs no puede viajar por las redes. Si se infectan varias máquinas en una red, que es porque el virus se extendió vía los discos flexibles de la manera usual. Así es como un virus de archivo se extiende en una red.

1. La computadora del usuario 1 se ve infectada, quizás por un disco de demostración de un vendedor. El virus queda residente en memoria (TSR).

2. El usuario 1 ejecuta otros programas en su disco duro. Ellos se infectan.

3. El usuario 1 ejecuta algunos programas en la red. Ellos se infectan. Una red emula un dispositivo de DOS; cuando se leen o escriben archivos en el servidor se hace exactamente de la misma manera como si fuera localmente. El virus no tiene que porqué comportarse de manera diferente para infectar archivos en el servidor.

4. El usuario 2 ingresa en el servidor, y ejecuta un archivo infectado. El virus es ahora TSR en la máquina del usuario 2.

5. El usuario 2 ejecuta otros programas, en su disco duro local, y en el servidor. Cada archivo se infecta.

6. Los usuarios 3, 4 y 5 ingresan al servidor y al ejecutar infectan archivos.

7. Y así sucesivamente.

PROTECCION DE REDES

Muchas redes usaban Novell NetWare, así que lo usaremos como ejemplo, pero usted puede adaptar los mismos principios para otros sistemas operativos de red.

1. Usted puede hacer directorios de sólo-lectura. Si usted hace archivos de sólo-lectura en el disco duro local, usted está perdiendo su tiempo, porque casi todos los virus de archivo les harán lectura-escritura, los infectarán, y los harán de nuevo de sólo-lectura. Esto es porque el usuario tiene el privilegio para hacer lectura-escritura los archivos en su disco duro local. Pero en un servidor de archivos, usted no tiene que darle ese privilegio al usuario, y el virus tiene el mismo privilegio que el usuario. De hecho, el virus es el usuario, y puede hacer nada más de lo que el usuario pueda. No hay magia sobre los virus; ellos están sujetos a los mismos condicionamientos que cualquier otro programa.

Desgraciadamente, algunos paquetes no pueden ejecutarse en subdirectorios de sólo-lectura, porque requieren escribir la configuración o archivos temporales, en el mismo directorio.

2. Usted puede hacer que un programa sea sólo-ejecutar. Esto significa que, aunque el directorio sea lectura-escritura, los ejecutables no pueden ni escribirse ni leerse. Ellos sólo pueden ejecutarse. Sea precavido, ya que en Novell NetWare ésta es un camino en un solo

sentido. Una vez que usted ha hecho un archivo sólo-ejecutar, no puede regresar. Todo lo que puede hacer es borrarlo, aun siendo Supervisor. Algunos programas no se ejecutarán son sólo-ejecutar, porque tienen archivos complementarios (OVL) que se encadenan en el extremo del archivo EXE. Así, si el archivo EXE no puede leerse, los complementos no pueden cargarse.

3. Usted puede hacer sólo-lectura archivos individuales usando el atributo de DOS, de esa manera se niega al usuario el privilegio de modificar atributos en ese directorio.

Usando una combinación de las tres técnicas mencionadas, usted debe poder hacer no-infectables un porcentaje grande de los archivos del servidor [de hecho, no-modificables sin la intervención del Supervisor]. Esto detiene virus que infectan la mayoría de los ejecutables en el servidor. El usuario tiene acceso de lectura-escritura a los datos [lo necesita para hacer su trabajo], y así es que el virus también tiene acceso de lectura-escritura a los datos. El borrado de archivos es lo que menos nos debe preocupar. Considere que en algunos virus la rutina de daño consiste en alterar archivos. Así, ¿cómo protegemos efectivamente los datos en el servidor?. La única respuesta es mantener los virus también fuera de las estaciones de trabajo.

La próxima cosa que usted puede hacer, si su servidor está ejecutando Novell NetWare, es ejecutar un anti-virus NLM [NetWare Loadable Module] en el servidor. Éste puede programarse para inspeccionar los archivos del servidor. El uso de un escáner de acceso basado en el servidor [como el File Access Monitor en el anti-virus Toolkit de Dr Solomon para NetWare] proporciona una defensa de niveles múltiples contra infecciones de virus, verificando los archivos cuando pasan a y desde las estaciones de trabajo. Además, puede negarse el acceso a los usuarios al servidor a menos que su estación de trabajo esté protegida.

Una protección similar está disponible para Windows NT. El anti-virus Toolkit de Dr Solomon para Windows NT ofrece protección completa, tanto para las estaciones de trabajo como para los servidores. FindVirus proporciona un escáner a pedido; y un programador de tareas, para verificar el sistema en momentos pre-definidos. Winguard para Windows NT proporciona protección constante, verificando archivos y discos antes de que se accedan.

Referencias: https://www.pandasecurity.com/es/security-info/malware/

6.5 AMENAZAS ACTUALES

En la actualidad se considera "**MALWARE**" tanto a virus, gusanos, troyanos, spyware, rootkits, exploits, adware, botnets, ransomware, etc. En tanto que los hoax basan su efecto en la ingeniería social.

6.5.1 Spyware

Cualquier aplicación que recopila la información sobre la actividad en el PC de una persona u organización tanto en línea como fuera de ésta y que es capaz de guardar los datos recopilados localmente o enviarlos a terceros sin que el objeto de espionaje se entere de ello. El spyware puede recopilar la información sobre los correos electrónicos y hasta sobre contraseñas y números de tarjetas de crédito.

6.5.2 Adware

Un programa que puede mostrar anuncios mientras se esté ejecutado o mediante otros mecanismos se llama adware. A menudo estos sistemas publicitarios se integran en los programas gratuitos ayudando a los desarrolladores a cubrir los gastos o les rinden ganancias. No obstante, con el tiempo el adware empezó a gozar de mala fama, ya que muchas veces estos programas no sólo descargan y muestran los anuncios, sino que envían información sobre el usuario sin consentimiento expreso de éste.

6.5.3 Rootkits

Se trata de programas diseñados para ocultar objetos como procesos, archivos o entradas del Registro de Windows. Este tipo de software no es malicioso en sí mismo, pero es utilizado por los creadores de malware para esconder evidencias y utilidades en los sistemas infectados. Existen ejemplares de malware que emplean rootkits con la finalidad de ocultar su presencia en el sistema en el que se instalan.

6.5.4 Exploits

Es una técnica o un programa que aprovecha un fallo o hueco de seguridad -una vulnerabilidad- existente en un determinado protocolo de comunicaciones, sistema operativo, o herramienta informática.

Dicho error se refiere a operaciones que provocan un funcionamiento anormal de la aplicación, y que pueden ser producidas intencionadamente por una persona con fines maliciosos, como la ejecución de código remoto, denegación de servicio, revelación de información o elevación de privilegios.

6.5.5 Gusanos

Los gusanos son en realidad una subclase de virus, por lo que comparten características. Son programas que realizan copias de sí mismos, alojándolas en diferentes ubicaciones de la computadora.

El objetivo de este malware suele ser colapsar las computadoras y las redes informáticas, impidiendo así el trabajo a los usuarios. A diferencia de los virus, los gusanos no infectan archivos.

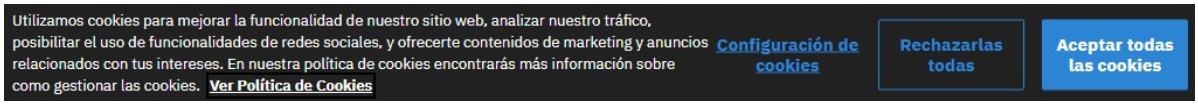
Los gusanos suelen utilizar técnicas de ingeniería social para conseguir mayor efectividad. Para ello, los creadores de malware seleccionan un tema o un nombre atractivo con el que camuflar el archivo malicioso. Los temas más recurrentes son los relacionados con el sexo, famosos, temas de actualidad o software pirata.

6.5.6 Cookies

Las cookies son pequeños archivos de texto que el navegador de Internet guarda en el ordenador del usuario cuando se visitan páginas web. La información que guardan las cookies se utiliza con distintos fines: para personalizar la navegación por la página, para recoger información demográfica sobre cuántos usuarios visitan la página y el tiempo que permanecen en ella, para realizar un seguimiento de los banners mostrados a cada usuario, entre otras cosas.

Por ejemplo, si un usuario visita una página web con cierta frecuencia, la cookie recuerda información sobre esa página, como nombre de usuario o contraseña de acceso facilitando así la navegación por la misma.

Aunque las cookies no son maliciosas como tal, un uso malintencionado de las mismas puede derivar en una amenaza para la privacidad del usuario, ya que pueden utilizarse para crear un perfil del usuario, con información que el usuario no controla y que puede ser enviada posteriormente a terceros.



Actualmente, las páginas Web que utilizan cookies deben solicitar la aprobación del visitante para hacer uso de ellas.

6.5.7 Hoax

Los “virus” HOAX se difunden por la llamada “ingeniería social”. En general a través de un mensaje de texto se le solicita que lo reenvíe a cuantas personas conozca con motivos más o menos verosímiles. En realidad, el virus es el mensaje, que de cumplir su cometido saturará los servidores de correo y generará tráfico innecesario en la red. Algunos mensajes son más directos en su objetivo de hacer daño y solicitan borrar archivos de sistema aduciendo que son virus, nuevamente aprovechando el desconocimiento de los usuarios y nombres de archivos “sospechosos”. Otras veces se usan para obtener direcciones de correo válidas que luego se podrán usar para enviar publicidad u otros fines. Lo recomendable es no reenviar este tipo de correos.

6.5.8 Virus de ejecución especulativa

Los virus Meltdown y Spectre se aprovechan de las características de ejecución especulativa de los procesadores, pero especialmente del procesador Intel, usando código que no debería haberse ejecutado pero que el procesador realiza como una de sus propiedades de mejora de la performance. Esto permite al virus acceder a memoria de kernel, a la que en modo usuario no debería hacerlo. Con la actualización del microcódigo se puede subsanar esta debilidad.

6.5.9 Ransomware

Un ransomware (del inglés ransom, «rescate» y ware, contracción de software), o «secuestro de datos» en español, es un tipo de programa dañino que restringe el acceso a determinadas partes o archivos del sistema operativo infectado y pide un rescate a cambio de quitar esta restricción.

Algunos tipos de ransomware cifran los archivos del sistema operativo inutilizando el dispositivo y coaccionando al usuario a pagar el rescate. Se han propuesto algunas alternativas en español al término en inglés, como "programa de secuestro", "secuestrador", «programa de chantaje» o «chantajista». En los primeros tiempos el pago se hacía a través de cuentas bancarias de países opacos, pero como el rastreo al delincuente era posible, se pasó al uso de monedas virtuales no rastreables.

6.5.10 Botnets

Son redes de computadoras con infección de malware que los cibercriminales utilizan para realizar tareas en línea sin el permiso del usuario. La palabra botnet es una combinación de robot y red (Net en inglés). Una botnet es un número de dispositivos conectados a Internet, cada uno de los cuales está ejecutando uno o más bots. El término se utiliza generalmente con una connotación negativa o maliciosa, derivada de usos criminales que se hacen de estas redes.

Un botnet no es un virus en sí mismo, sino una es una colección de dispositivos automáticos conectados. Si estos han sido infectados con un malware, los cibercriminales pueden utilizarlos para obtener el control y distribuir programas nocivos.

Los atacantes suelen utilizar sus botnets para lanzar ataques DDoS, así como para enviar SPAM en forma de correos electrónicos no deseados, detectar contraseñas confidenciales o distribuir ransomware.

6.5.11 Phishing

El phishing es cuando los cibercriminales atacan a sus víctimas con correos electrónicos que parecen ser de una empresa legítima que solicita información confidencial. Los ataques de

phishing se utilizan a menudo para inducir a que las personas entreguen sus datos de tarjetas de crédito y otra información personal.

6.5.12 Inyección de código SQL

Una inyección de código SQL (por sus siglas en inglés Structured Query Language) es un tipo de ciberataque utilizado para tomar el control y robar datos de una base de datos. Los cibercriminales aprovechan las vulnerabilidades de las aplicaciones basadas en datos para insertar código malicioso en una base de datos mediante una instrucción SQL maliciosa. Esto le brinda acceso a la información confidencial contenida en la base de datos.

6.5.13 "Man-in-the-middle"

Un ataque de tipo "Man-in-the-middle" es un tipo de ciberamenaza en la que un cibercriminal intercepta la comunicación entre dos individuos para robar datos. Por ejemplo, en una red Wi-Fi no segura, un atacante podría interceptar los datos que se transmiten desde el dispositivo de la víctima y la red.

6.5.14 Denegación de servicio

Un ataque de denegación de servicio es cuando los cibercriminales impiden que un sistema informático satisfaga solicitudes legítimas sobrecargando las redes y los servidores con tráfico. Esto hace que el sistema sea inutilizable e impide que una organización realice funciones vitales.

6.5.15 Backdoor

Los virus de puerta trasera son aquellos en los que los hackers acceden a funciones de la computadora de manera oculta y trabajan en segundo plano. Este tipo de virus son, en realidad, una combinación de diferentes amenazas de seguridad y algunos funcionan automáticamente y no necesitan ser controlados de manera remota.

A menudo, las puertas traseras se introducen en el sistema gracias a otros programas nocivos como troyanos, virus o incluso spyware. Consiguen acceso sin que el administrador se dé cuenta y luego infectan las sesiones a cada uno de los usuarios conectados a la red comprometida. Es posible que algunas amenazas puedan ser colocadas con carácter previo por usuarios que tienen los privilegios adecuados, para ganar acceso más tarde.

6.5.16 Keylogger

El término keylogger se refiere a un programa informático malicioso que registra en secreto todas las pulsaciones sobre el teclado realizadas por el usuario de un ordenador. Los Keyloggers se utilizan para obtener acceso ilegal a información confidencial, como por ejemplo datos personales, datos de tarjetas de crédito, credenciales de acceso, etc.

6.6 AMENAZAS INFORMATICAS DEL FUTURO

Si en un momento el objetivo de los ataques fue cambiar las plataformas tecnológicas, ahora las tendencias **cibercriminales** indican que la nueva modalidad es manipular los certificados que contienen la información digital. El área semántica, era reservada para los humanos, se convirtió ahora en el núcleo de los ataques debido a la evolución de la Web 2.0 y las redes sociales, factores que llevaron al nacimiento de la generación 3.0.

- Se puede afirmar que "la Web 3.0 otorga contenidos y significados de manera tal que pueden ser comprendidos por las computadoras, las cuales -por medio de técnicas de inteligencia

artificial- son capaces de emular y mejorar la obtención de conocimiento, hasta el momento reservada a las personas".

- Es decir, se trata de dotar de significado a las páginas web, y de ahí el nombre de Web semántica o Sociedad del Conocimiento, como evolución de la ya pasada Sociedad de la Información.

En este sentido, las amenazas informáticas que viene en el futuro ya no son con la inclusión de troyanos en los sistemas o softwares espías, sino con el hecho de que los ataques se han profesionalizado y manipulan el significado del contenido virtual.

- "La Web 3.0, basada en conceptos como elaborar, compartir y significar, está representando un desafío para los **hackers** que ya no utilizan las plataformas convencionales de ataque, sino que optan por modificar los significados del contenido digital, provocando así la confusión lógica del usuario y permitiendo de este modo la intrusión en los sistemas", La amenaza ya no solicita la clave de homebanking del desprevenido usuario, sino que directamente modifica el balance de la cuenta, asustando al internauta y, a partir de allí, sí efectuar el robo del capital".
- Obtención de perfiles de los usuarios por medios, en un principio, lícitos: seguimiento de las búsquedas realizadas, históricos de navegación, seguimiento con geoposicionamiento de los móviles, análisis de las imágenes digitales subidas a Internet, etc.

Referencias: https://en.wikipedia.org/wiki/Computer_security, https://www.universidadviu.com/que-es-la-seguridad-informatica; https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security

6.7 CIFRADO DE LA INFORMACIÓN O CRIPTOGRAFÍA

La criptografía es la técnica de escribir de manera secreta u oculta, buscando proteger la información.

Para que un mensaje que, desde que es generado por el emisor, deba atravesar un medio de comunicación inseguro para luego llegar al receptor, desde hace muchos años se emplean métodos para mantenerlo oculto.

6.7.1 Cifrado simétrico

El cifrado simétrico consiste en la codificación mediante una regla (clave) que sólo conocen el emisor y el receptor.

Una vez de acuerdo, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra usando la misma clave.

El inconveniente de este método es que se debe enviar al receptor la clave de cifrado con algún método seguro y si el receptor se encuentra lejos esto puede ser costoso.

Como ejemplo de sistema simétrico está «Enigma»; Éste fue un sistema usado por Alemania en la segunda guerra mundial, en el que las claves se distribuían a diario en forma de libros de códigos. Cada día, un operador de radio, receptor o transmisor consultaba su copia del libro de códigos para encontrar la clave del día. Todo el tráfico enviado por ondas de radio durante aquel día era cifrado y descifrado usando las claves del día.

En la Web de TDIII hay dos videos muy interesantes sobre el código Enigma, que recomendamos ver.

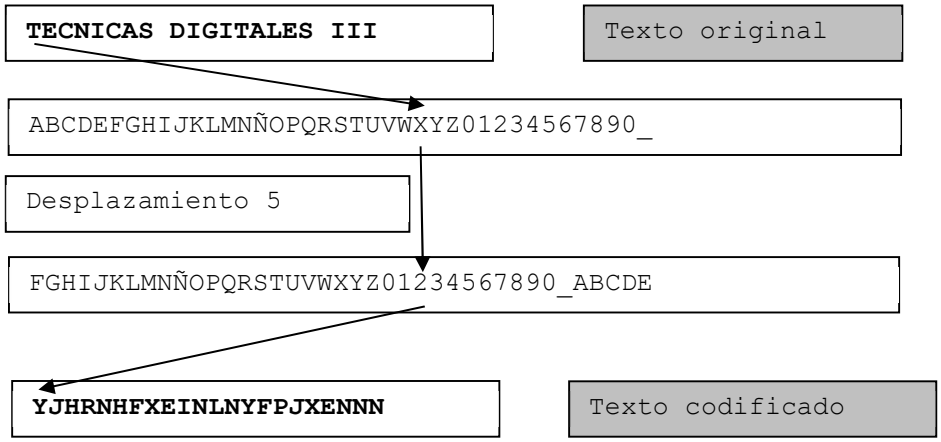


Figura 6.7.1 – Cifrado simétrico

Se analizará un ejemplo de codificación simple:

El texto original es "**TECNICAS DIGITALES III**" y el método de cifrado es incrementar en 5 la posición de cada letra. Disponemos de una lista de letras ordenadas y luego la misma lista, pero desplazada 5 posiciones. Para la primera letra (T) corresponde (Y), luego para la segunda letra (E) corresponde (J) y así sucesivamente.

Se obtiene entonces el texto codificado "**YJHRNHFXEINLNYFPJXENNN**", el que por supuesto es incomprensible.

Con el proceso inverso, conocido por el receptor, se puede reconstruir el texto original.

6.7.2 Cifrado asimétrico

Este método, atribuido a Diffie y Hellman, consiste en la generación matemática de dos claves. Una de ellas, llamada privada, es conocida sólo por el emisor y la otra, llamada pública es enviada al receptor. Los textos codificados con la clave privada pueden luego decodificarse con la clave pública. Esto facilita la distribución de claves en los sistemas de comunicación actuales y da lugar a la PKI (Public Key Infrastructure).

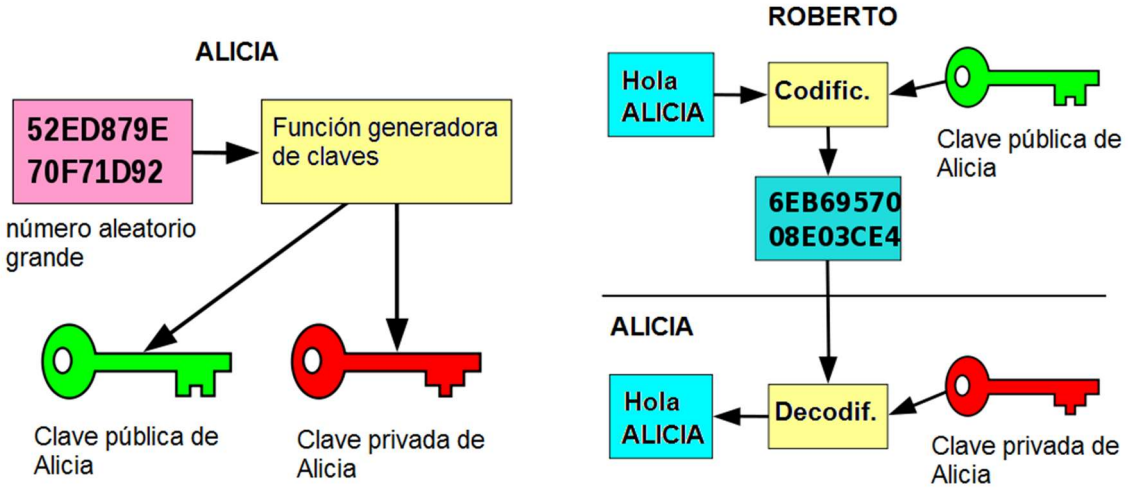


Figura 6.7.2 – Codificación asimétrica – generación de claves y envío de mensajes

6.7.3 Cifrado híbrido

Un sistema de cifrado híbrido usa tanto el cifrado simétrico como el asimétrico. Funciona mediante el uso de una clave pública para compartir una clave privada para el cifrado simétrico. El mensaje que se esté enviando en el momento, se cifra usando la clave y enviándolo al destinatario. Ya que compartir una clave simétrica no es seguro, la clave usada es diferente para cada sesión. La clave de sesión es cifrada con la clave pública, y el mensaje saliente es cifrado con la clave simétrica, todo combinado automáticamente en un sólo paquete. El destinatario usa su clave privada para descifrar la clave de sesión y acto seguido usa la clave de sesión para descifrar el mensaje.

Nota: El tema tiene una complejidad que excede el alcance de nuestra asignatura y de este apunte. Si desea más información, se cita un excelente trabajo de tesis en: https://repository.unad.edu.co/bitstream/handle/10596/40365/rtorrescar.pdf?sequence=3&isAllowed=y.

También se puede consultar el artículo "GNU privacy guard" en https://www.gnupg.org/gph/es/manual/c190.html

6.8 MÓDULO DE PLATAFORMA CONFIABLE

Trusted Platform Module (TPM, también conocido como ISO /IEC 11889) es un estándar internacional para un criptoprocesador seguro, un microcontrolador dedicado diseñado para proteger el hardware a través de claves criptográficas integradas. El término también puede referirse a un chip que cumple con el estándar.

Trusted Platform Module proporciona:

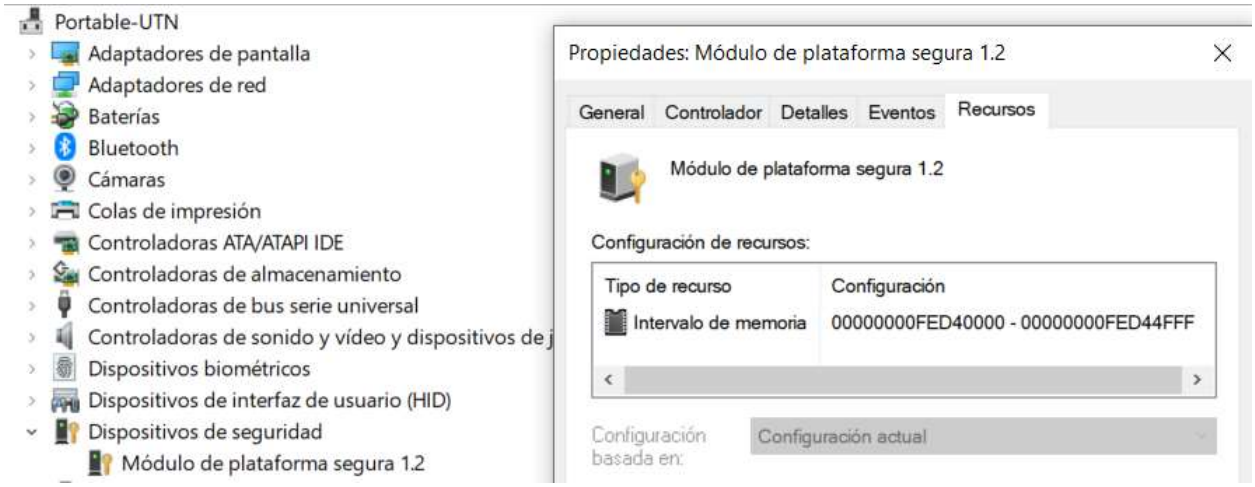
- Un generador de números aleatorios por hardware
- Facilidades para la generación segura de claves criptográficas para usos limitados.
- Atestación remota: crea un resumen de clave hash casi indescifrable de la configuración de hardware y software. El software encargado de "hashear" los datos de configuración determina el alcance del resumen. Esto permite a un tercero verificar que el software no se ha cambiado.
- Enlace: Cifra los datos utilizando la clave de enlace TPM, una clave RSA única descendiente de una clave de almacenamiento.
- Sellado: Similar al enlace, pero, además especifica el estado TPM para los datos que se descifrará.
- Otras funciones de Trusted Computing (Computación confiable) para que los datos se descifren.

Los programas informáticos pueden usar un TPM para autenticar dispositivos de hardware, ya que cada chip TPM tiene una clave de aprobación (EK) única y secreta que se graba a medida que se produce. La seguridad integrada en el hardware proporciona más protección que una solución solo de software.

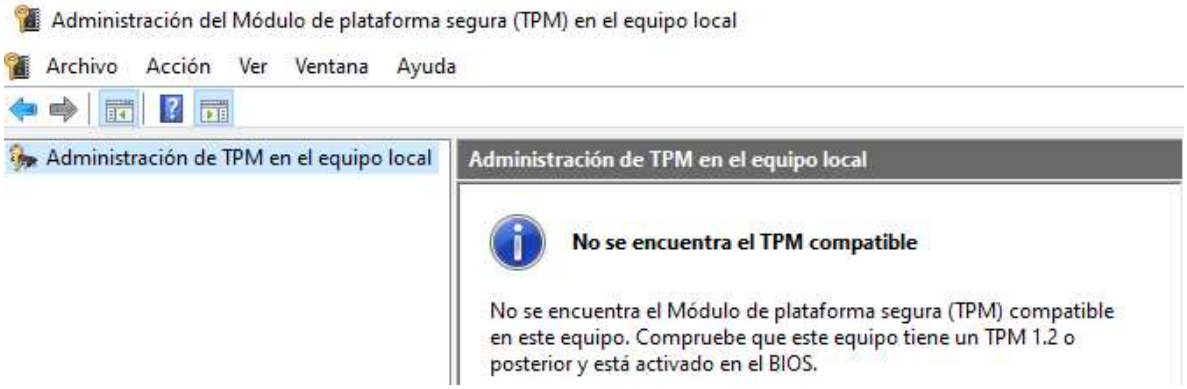
El Departamento de Defensa de los Estados Unidos (DoD) especifica que "los nuevos activos informáticos (por ejemplo, servidor, computadora de escritorio, computadora portátil, cliente ligero, tableta, teléfono inteligente, asistente digital personal, teléfono móvil) adquiridos para admitir el DoD incluirán una versión 1.2 o superior de TPM cuando lo requieran las Guías de Implementación Técnica de Seguridad (SURG) de la Agencia de Sistemas de Información de Defensa (DISA) y donde dicha tecnología esté disponible". El Departamento de Seguridad anticipa que TPM se utilizará para la identificación, autenticación, cifrado y verificación de la integridad del dispositivo.

La nueva versión de Windows, Windows 11, exige que el dispositivo donde será instalado posea al menos TPM 1.2.

En una PC con esta característica, el administrador de dispositivos de Windows muestra un aspecto similar al de la figura siguiente.



Para verificar si su PC tiene habilitado el módulo TPM, se debe ejecutar el comando de consola tpm.msc. En este caso, TPM está deshabilitado o no existe.



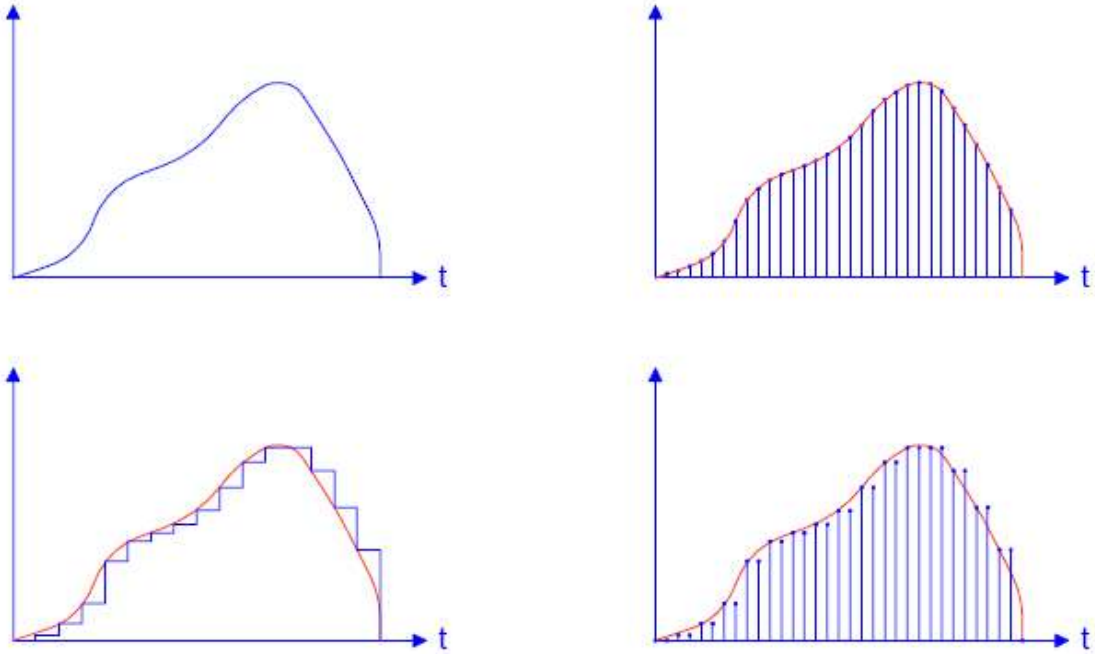
CAPÍTULO 7

PROCESAMIENTO DIGITAL DE SEÑALES

7.1 SEÑALES

Se tratarán 4 tipos de señales:

- Analógicas, x(t): amplitud y tiempo continuos.
- Muestreadas, X[n], tiempo discreto, amplitud continua.
- Cuantizada, Xq[t], tiempo continuo, amplitud discreta.
- Digital, -xq[n], amplitud y tiempo discretos.



Clasificación de las señales

- Según su duración:
  - Continuas: Se definen para todo tiempo *t*.
  - Periódicas: $xp(t) = xp(t \pm nT)$, donde *T* es el periodo y *n* es un entero.
  - Causales: Son 0 para $t<0$. Se definen sólo para el eje positivo de *t*.
  - Anticausales: Son 0 para $t>0$. Se definen sólo para el eje negativo de *t*.
  - No causales: Se definen para ambos ejes de *t*.

- Basada en la simetría
  - Simetría Par: $x(t) = x(-t)$
  - Simetría Impar: $x(t) = -x(-t)$

- En energía y potencia (impulsos limitados en tiempo y señales periódicas)
  - Energía de una señal:

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 \, dt$$

- Potencia de una señal:

$$P_x = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{T_0} |x(t)|^2 \, dt$$

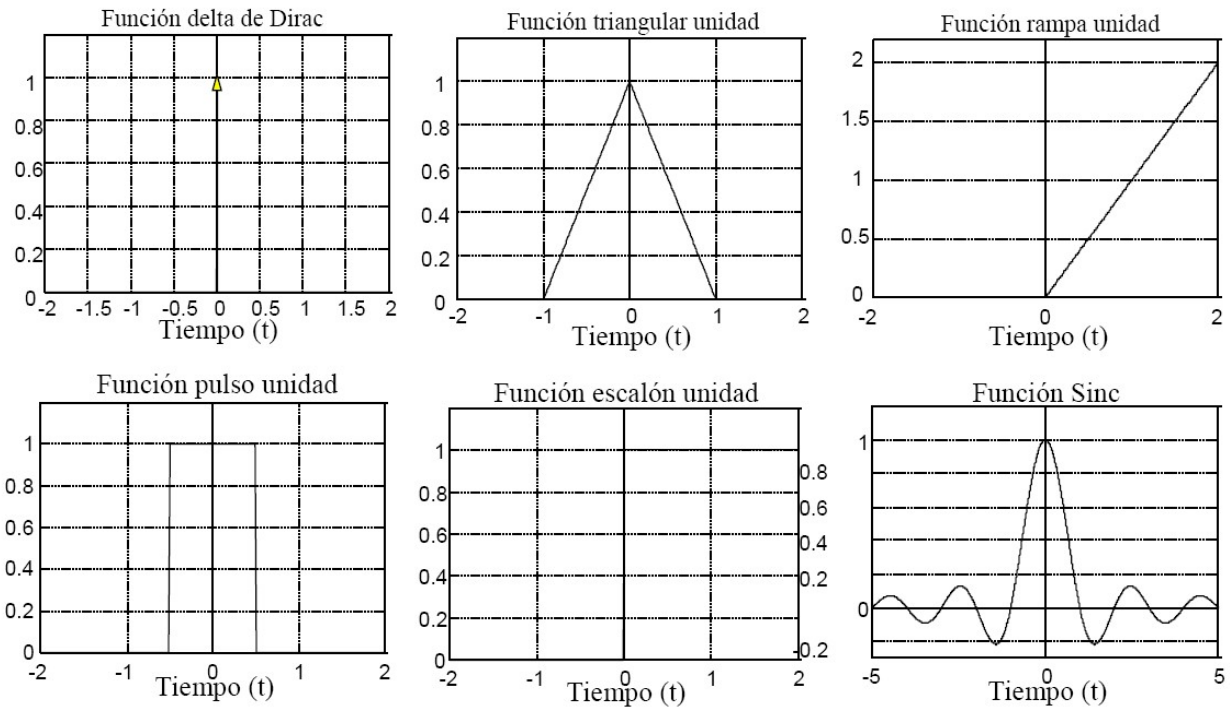
Una señal se dice que es de energía si *Ex* es finito, lo que implica que *Px* es 0. Ej. Pulsos limitados en el tiempo.

Una señal se dice que es de potencia si *Px* es finito, lo que implica que *Ex* es infinito. Ej. Una señal periódica.

Funciones elementales:

- Escalón unidad : *u(t)*
- Rampa : *r(t)=t u(t)*
- Pulso : *u(t+1/2)-u(t-1/2)*
- Triangular : *tri(t)=r(t+1)-2r(t)+r(t-1)*
- Seno Cardinal , Sinc: *sinc(t)= sen (πt)/*πt
- Impulso δ(t) o función delta de Dirac: $\delta(t) = 0$, $t \neq 0$; $\int_{-\infty}^{+\infty} \delta(\tau)\, d\tau = 1$

Representación de las señales:



Operaciones con señales:

- Desplazamiento en el tiempo: x(t-2), desp. A la derecha
- Compresión en el tiempo: x(2t)
- Dilatación en el tiempo: x(t/2)
- Reflexión: x(-t)

Algunas señales en Matlab

```
>> y = diric (x,N)
```

La función de Dirichlet se define de la siguiente forma:

$D(x) = \sin(Nx/2) / N \sin(x/2)$

El argumento de entrada es un vector x en cuyos puntos queremos calcular la función de dirichlet y el parámetro N, e es el número de máximos de la función en el intervalo (0-2π).

```
>> y = sawtooth (x,width)
```

Genera una señal en diente de sierra con período 2π para los elementos del vector x. El parámetro "width" es un escalar entre 0 y 1 y describe la fracción del período 2π en el que ocurre el máximo.

```
>> y = sinc (x)
```

La función sinc (x) = sin (πx) / (πx)

```
>> y = square (x, duty)
```

Genera una onda cuadrada de período 2π con un ciclo de trabajo dado. El parámetro "duty" es el porcentaje del período en el cual la señal es positiva.

7.2 SISTEMAS

- Un sistema físico es un conjunto de dispositivos conectados entre sí, cuyo funcionamiento está sujeto a leyes físicas.
- Para nosotros un sistema es un procesador de señales.
- Las señales a ser procesadas son la excitación del sistema.
- La salida del sistema es nuestra señal procesada.
- El sistema se representa mediante ecuaciones diferenciales que relacionan la salida y(t) y la entrada x(t)mediante constantes, parámetros y variables independientes.

Sistemas: Clasificación

Los sistemas se clasifican en:

- Lineales: los coeficientes no dependen de x o y, no hay términos constantes.
- No lineales: los coeficientes dependen de x o y, hay términos constantes.
- Invariante en el tiempo: Los coeficientes no dependen de t.
- Variante en el tiempo: Los coeficientes son funciones de t.

A los sistemas lineales se les puede aplicar el principio de superposición.

Si $x(t)=x1(t)+x2(t) \rightarrow y(t)= y1(t)+y2(t)$
$x(t)=K\ x1(t) \rightarrow y(t)=K.\ y1(t)$

Un sistema es invariante en el tiempo cuando la respuesta y(t) depende sólo de la forma de la entrada x(t) y no del tiempo en que se aplica. Matemáticamente:

Si L{x(t)}=y(t) -> L{x(t-t0)}=y(t-t0)
L{} indica el sistema físico en cuestión.

Usaremos sistemas LTI: lineal e invariante en el tiempo. La respuesta al impulso del sistema se representa con h(t) y es la respuesta a la excitación delta de Dirac.

Es la principal herramienta para el estudio de un sistema.

7.3 CONVOLUCIÓN

Podremos calcular la respuesta y(t) de un sistema a una entrada cualquiera x(t).

Condiciones para llevarla a cabo:

- Sistema LTI
- Respuesta al impulso del sistema h(t)

Basándonos en el principio de superposición y en que el sistema es invariante en el tiempo:

$$Si\ T\{\delta(t)\} = h(t) \rightarrow T\{K \cdot \delta(t - t_0)\} = K \cdot h(t - t_0)$$

Una señal arbitraria de entrada *x(t)* puede expresarse como un tren infinito de impulsos. Para ello, dividimos *x(t)* en tiras rectangulares de anchura *ts* y altura *x(k ts)*. Cada tira la reemplazamos por un impulso cuya amplitud es el área de la tira:

ts . x(k.ts) δ(t –kts)

La función xs(t) que aproxima x(t) es: $x_s(t) = \sum_{-\infty}^{\infty} t_s x(kt_s).\delta(t - kt_s)$

x(t) es el límite cuando ts -> dλ , kts->λ $x(t) = \lim_{ts \to 0} \sum_{k=-\infty}^{k=\infty} t_s x(kt_s).\delta(t - kt_s) = \int x(\lambda).\delta(t - \lambda).d\lambda$

Y aplicando el principio de superposición:

$$y(t) = T\{x(t)\} = T\left\{\int_{-\infty}^{\infty} x(\lambda) \cdot \delta(t - \lambda) \cdot d\lambda\right\} = \int_{-\infty}^{\infty} x(\lambda) \cdot T\{\delta(t - \lambda)\} \cdot d\lambda = \int_{-\infty}^{\infty} x(\lambda) \cdot h(t - \lambda) \cdot d\lambda = x(t) * h(t)$$

Mediante convolución hemos sido capaces de determinar la respuesta del sistema a una señal de entrada a partir de la respuesta del sistema a una entrada impulso.

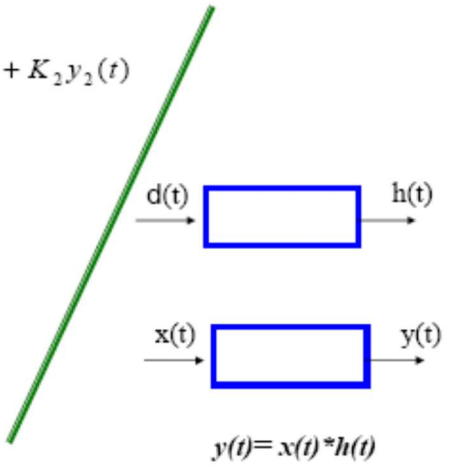
La función h(t) se define para t >= 0 y decrece cuando t -> 0, para la mayoría de los sistemas físicos. Por tanto:

- La respuesta en t_0 depende de los valores actuales y pasados de la entrada y de la respuesta al impulso.
- Los valores más recientes de x(t) son multiplicados por sus correspondientes más antiguos (y más grandes) valores de h(t).

Propiedades de la convolución:

- **Supóngase que** $x(t)*h(t)=y(t)$) **entonces:**

$$[x_1(t)+x_2(t)]*h(t)=y_1(t)+y_2(t)$$
$$[K_1x_1(t)+K_2x_2(t)]*h(t)=K_1y_1(t)+K_2y_2(t)$$
$$x(t)*h(t-\alpha)=y(t-\alpha)$$
$$x(t-\alpha)*h(t-\beta)=y(t-\alpha-\beta)$$
$$\delta(t)*h(t)=h(t)$$
$$x(t)*h'(t)=x'(t)*h(t)=y'(t)$$
$$x'(t)*h'(t)=y''(t)$$
$$x^m(t)*h^n(t)=y^{m+n}(t)$$
$$x(\alpha t)*h(\alpha t)=\left|\frac{1}{\alpha}\right|y(\alpha t)$$



7.4 CORRELACIÓN

Es una operación similar a la convolución, con la diferencia de que en la correlación no se refleja una de las señales.

$$R_{xy}(t)=x(t)**y(t)=\int_{-\infty}^{\infty}x(\lambda)y(\lambda-t)d\lambda=x(t)*y(-t)$$

- La correlación nos da una medida de la similitud entre dos señales.
- No existe la propiedad conmutativa por lo que dadas dos señales x(t) e y(t) se definen dos correlaciones:

$$R_{xy}(t)=x(t)**y(t)=\int_{-\infty}^{\infty}x(\tau)y(\tau-t)d\tau \qquad R_{xy}[n]=\sum_{k=-\infty}^{\infty}x[k]y[k-n]$$

$$R_{yx}(t)=y(t)**x(t)=\int_{-\infty}^{\infty}y(\tau)x(\tau-t)d\tau \qquad R_{yx}[n]=\sum_{k=-\infty}^{\infty}y[k]x[k-n]$$

Que solo coinciden en t=0: $R_{xy}(0)=R_{yx}(0)$

Nota: correlación discreta

$$R_{xy}[n]=\sum_{k=-\infty}^{\infty}x[k]y[k-n] \quad \text{para } n=0,\pm1,\pm2,\pm3,\cdots$$

7.4.1 Autocorrelación

La correlación de una señal consigo misma se denomina autocorrelación:

$$R_{xx}(t)=x(t)**x(t)=\int_{-\infty}^{\infty}x(\lambda)x(\lambda-t)d\lambda$$

La autocorrelación representa la similitud entre la señal y su desplazada. El máximo de autocorrelación se obtiene cuando no hay desplazamiento (t=0). La autocorrelación es simétrica con respecto al origen, ya que $R_{xx}(t) = R_{xx}(-t)$

Nota: Autocorrelación discreta

$$R_{xx}[n] = \sum_{k=-\infty}^{\infty} x[k]x[k-n] \quad \text{para } n = 0, \pm 1, \pm 2, \pm 3, \cdots$$

Ejemplo de uso de la autocorrelación: Radar.

7.5 CONVOLUCIÓN DISCRETA

Cuando se trata de hacer un procesamiento digital de Cuando se trata de hacer un procesamiento digital de señal no tiene sentido hablar de convoluciones aplicando estrictamente la definición ya

que sólo disponemos de valores en instantes discretos de tiempo. Es necesario, pues, una aproximación numérica.

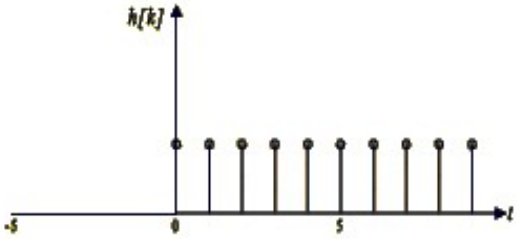
Para realizar la convolución entre dos señales, se evaluará el área de la función x(l)h(t-l). Para ello, disponemos de muestreos de ambas señales en los instantes de tiempo nts, que llamaremos xs[k] y hs[n-k]

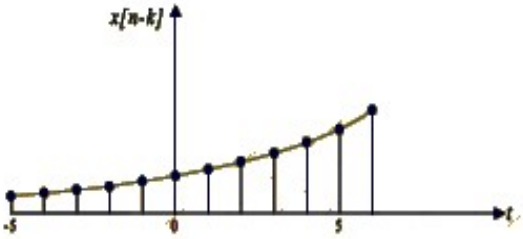
(donde n y k son enteros). El área es, por tanto,

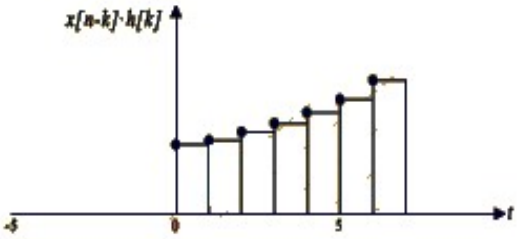
$$y_s[n] = \sum_{k=-\infty}^{\infty} t_s . x_s[k] . h_s[n-k] = \sum_{k=-\infty}^{\infty} x_s[k] . h_s[n-k]$$

La convolución discreta se define para un intervalo de muestreo $t_s = 1$

$$y_s[n] = x[n] * . h[n] = \sum_{k=-\infty}^{\infty} x_s[k] . h_s[n-k]$$







A veces es posible hacer una convolución discreta analítica. Veamos un ejemplo. Se trata de hacer la convolución de una señal x[n]=nu[n+1] con h[n]=a^{-n}u[n], siendo a<1.

$$y[n]=\sum_{k=-\infty}^{\infty} k.u[k+1].a^{-(n-k)}.u[n-k]=\sum_{k=-1}^{n} k.a^{-(n-k)}=a^{-(n-k)}+a^{-n}\sum_{k=0}^{n} k.a^{k}=$$
$$=a^{-(n+1)}+a^{-n}.(a+2a^2+3a^3+...+n.a^n)$$
$$=a^{-(n+1)}+a^{-n}.a(1+2a+3a^2+...+n.a^n)$$
$$=a^{-(n+1)}+\frac{a^{-n+1}}{(1-a)^2}.(1-(n+1)a^2+na^{n-1})$$

En la práctica se trabaja con secuencias de longitud finita. Para hacer la convolución, una de las secuencias se refleja y se desplaza sucesivamente. Veremos algunos métodos para calcular la convolución a partir de dos secuencias.

7.5.1 Propiedades sobre la duración de la convolución discreta.

El índice del comienzo de la convolución es la suma de los índices de comienzo de las respectivas señales. Si las dos señales comienzan en n=n0 y n=n1, la convolución

Para dos secuencias de duración M y N, su convolución se extiende durante M+N-1 muestreos.

- comienza en n=n0+n1.

Propiedades de la convolución discreta (x[n]*h[n]=y[n])

$y[n]=\sum x[k]\,h[n-k]$

con k variando entre ∞ *y* $-\infty$.

$[Ax_1 + Bx_2]*h = y_1 + y_2$

$x[n]*h[n-\alpha] = x[n-\alpha]*h[n] = y[n-\alpha]$

$x[n-\alpha]*h[n-\beta] = y[n-\alpha-\beta]$

$\delta[n]*h[n] = h[n]$

$h[n] = \delta[n]*h[n] = \{u[n]-u[n-1]\}*h[n] = y_u[n]-y_u[n-1]$
$u[n]*h[n] = \sum x[k]$
con k variando entre ∞ *y* $-\infty$.

$\{x[n]-x[n-1]\}*h[n] = y[n]-y[n-1]$

7.5.2 Convolución y Correlación en MATLAB

```
>> y = conv(x,h)
```

Hace la convolución de los vectores x y h. El vector resultante y tiene un tamaño igual a length(x)+length(h)-1

```
>> rxy = xcorr(x,y)
```

Hace la correlación de los vectores de M elementos x e y. Devuelve un vector de 2M-1 elementos.

```
>> rxx = xcorr(x)
```

Hace la autocorrelación del vector x de M elementos. Devuelve un vector de 2M-1 elementos.

7.6 SERIES Y TRANSFORMADA DE FOURIER

7.6.1 Series de Fourier

Las series de Fourier describen señales periódicas como una combinación de señales armónicas (sinusoides).

Se puede analizar una señal periódica en términos de su contenido frecuencial o espectro.

Dualidad entre tiempo y frecuencia. Forma trigonométrica de las series de Fourier: se pretende describir una función periódica x(t) de período T, frecuencia fundamental $f=1/T$, $\omega_0 = 2\pi f_0$.

$$x_p(t) = \frac{a_0}{2} + a_1 \cos(\omega_0 t) + ... + a_i \cos(k\omega_0 t) + ... + b_1 \sin(\omega_0 t) + ... + b_1 \sin(k\omega_0 t) =$$

$$= \frac{a_0}{2} \sum_{k=1}^{n} a_i \cos(k\omega_0 t) + b_1 \sin(k\omega_0 t)$$

7.6.2 Efecto Gibbs

Para señales discontinuas, su reconstrucción a partir de las series de Fourier produce el llamado efecto Gibbs, que consiste en la aparición de un pico de del 9% en el punto de discontinuidad. Aún se tiene este efecto cuando se utilicen gran cantidad de armónicos para la reconstrucción.

Al querer aproximar la función periódica que tiene infinitos armónicos hay que truncar la función hasta el armónico N -> se produce este efecto.

Para eliminarlo se usan las llamadas ventanas espectrales que suavizan la reconstrucción de la función.

7.6.3 Transformada de Fourier

Para ampliar el concepto de series de Fourier a señales no periódicas se puede visualizar una señal no periódica como una señal continua de período infinito.

- El espaciado entre frecuencias se aprox. A cero y es por lo tanto una función continua
- La señal pasa a ser de potencia a señal de energía.
- Los coeficientes Xs[k] son cero. Ya no es un indicador del contenido espectral de la señal.

Se define la Transformada de Fourier de x(t) como:

$$X(f) = \lim_{t \to \infty} T.X_S[k] = \int_{-\infty}^{\infty} x(t).\exp(-j2\pi ft)dt$$

7.6.4 Relación entre series y transformada de Fourier

X(w) es la función envolvente de Xs[k]

Si muestreamos X(w) a intervalos f_0. la función resultante es el espectro de una señal periódica de período $T_0=1/f_0$

Es decir, muestrear en el dominio frecuencial se corresponde con señales periódicas en el dominio temporal.

$$X(f) = T.X_S[k]|_{\text{kf0-f}}$$

$$X_S[k] = \frac{X(f)}{T}|_{\text{f- kf0}}$$

La transformada inversa de Fourier de X(w)

$$x(t) = \int_{-\infty}^{\infty} X(f).\exp(j2\pi ft).df$$

Podemos utilizar la Transformada de Fourier para analizar la respuesta a sistemas LTI, valiéndonos del hecho de que convolución en el tiempo equivale al producto en el dominio

frecuencial.

Si la respuesta y(t) a un sistema con una respuesta a impulso h(t) y entrada x(t) con condiciones iniciales cero es:

$$y(t)=x(t)*h(t)$$

Aplicando la Transformada de Fourier a ambos miembros,

$$Y(w)= X(\omega)H(\omega)$$

H(w)=Y(w)/X(w) es la función de Transferencia del sistema. Esta nos permite analizar la respuesta frecuencial del sistema.

Como se vio en las Series de Fourier, se puede analizar la respuesta en el estado estacionario del sistema a partir de H(w).

Limitaciones de la Transformada de Fourier:

- El sistema debe tener condiciones iniciales cero.
- Entradas que no son señales de energía requieren el uso de impulsos.

Por ello se extiende el concepto de la Transformada de Fourier a la Transformada de Laplace.

7.7 TRANSFORMADA DE LAPLACE

Se define la Transformada de Laplace de la señal x(t):

$$X(s)= \mathrm{L}\{x(t)\}= \int x(t)\exp(-st)dt$$

La cantidad compleja s=s+jw. De esta forma se generaliza el concepto de frecuencia en la Transformada de Fourier.

Se hace notar que el límite inferior de la integral es 0, lo cual proporciona

una misma transformada para señales causales ya que x(t) y

x(t).u(t) son iguales.

La Transformada de Laplace existe si la integral que la define es finita. Para ello se necesita que los valores de s sean unos concretos, lo que define una región de convergencia de la Transformada de Laplace.

Con la Transformada de Laplace se generaliza el concepto de función de Transferencia de un sistema a aquellos cuyas condiciones iniciales son no nulas.

$X(s)=L\{x(t)\} = \int x(t)\exp(-st)dt$

7.8 MUESTREO Y CUANTIZACIÓN

- El muestreo digital de una señal analógica trae consigo una discretización tanto en el dominio temporal como en el de la amplitud.
- Para describir matemáticamente el muestreo nos basaremos en el muestreo ideal. Consiste en una función que toma los valores de la señal Xc(t) en los instantes de muestreo y cero en los otros puntos.

$$x_s(t) = x_c(t) \sum \delta(t - nt_s) = \sum x_c(n\, t_s)\delta(t - nt_s) = x_c(t).\, x_I(t)$$

En $\sum$ la variación es de n entre ∞ y $-\infty$.

Donde ts es el período de muestreo y x(t) es la función de interpolación.

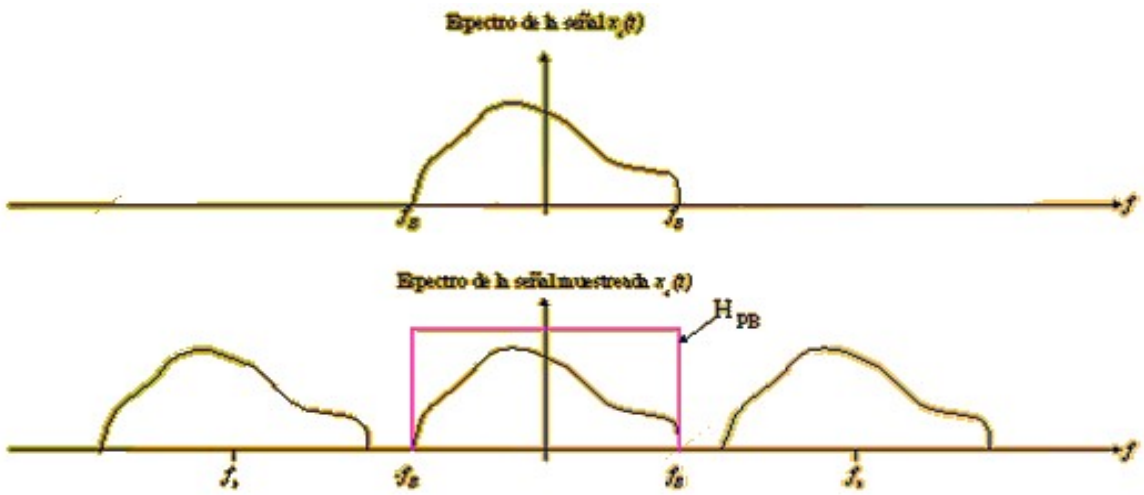
El muestreo trae aparejado pérdida de información de la señal original. El teorema del muestreo establece en qué condiciones se debe muestrear para que no se nos escapen los eventos de la señal original que son importantes para nuestro posterior desarrollo con la señal.

7.8.1 Teorema del muestreo

Una señal Xc(t) con un espectro limitado a la frecuencia Fb (|f|<=Fb) puede ser muestreada sin pérdida de información si la frecuencia de muestreo fs supera la cantidad 2Fb, es decir fs>=2Fb.

De no muestrearse al menos a esa frecuencia tiene lugar el fenómeno de "Aliasing".

Es decir, el espectro de la señal muestreada se compone de una función de período 1/t, replicándose en cada período el espectro de la señal original. En la sig. Fig. se observa el fenómeno:



Para recuperar la señal original a partir de la muestreada no tenemos más que aplicar un filtro pasa bajo con una frecuencia de corte en f=fB y una amplificación ts, es decir,

$X_C(f) = X_S(f).\ H_{PB}(f) \rightarrow x_C(t) = x_S(t) * h_{PB}(t)$

$H_{PB}(f) = t_s\ .rect(f/2f_B) \rightarrow h_{PB}(t) = 2.t_s\ .\ f_B\ .\ sinc(t.2.f_B)$
$x_S(t) = \sum x_C(k\ t_S).\ \delta(t - k\ t_S) = \sum x_C(k).\ \delta(t - k\ t_S)$

$x_C(t) = \sum x_C(k).\ h_{PB}(t - k\ t_S) =$
$= 2.t_S.f_B \sum x_C(k).\ sinc\ [2.f_B\ (t - k\ t_S)]$
con k variando entre ∞ y –∞.

A la función sinc(t) se le denomina función de interpolación cardinal.

Este tipo de reconstrucción tiene los siguientes problemas:

- El dominio de la función sinc es infinito
- Requiere muestreos pasados y futuros
- Se puede truncar la función sinc(t) -> aparecería el efecto Gibbs
- No es posible reconstruir funciones con discontinuidades.

7.8.2 Cuantización

Para procesar señales digitales no solo alcanza con muestrear la señal analógica, sino también cuantizar la amplitud de la señal a un número finito de niveles.

El tipo más usual es la cuantización uniforme en el que los niveles son todos iguales. La mayoría usan un número de niveles que es potencia de 2. Si $L = 2^b$, cada uno de los niveles es codificado a un número binario de b bits.

Ruido de Cuantización: Llamaremos Xs[n] a la señal discreta y Xq[n] a la señal discreta cuantizada. El error es:

$\varepsilon[n] = x_s[n] - x_Q[n]$

Se define la relación señal a ruido de cuantización (SNRQ) como la relación entre la potencia PS de la señal y la potencia PN del error e[n], medido en decibelios.

$$P_S = \frac{1}{N}.\sum_{n=-\infty}^{\infty} x_S^2[n] \qquad P_N = \frac{1}{N}.\sum_{n=-\infty}^{\infty} \xi^2[n]$$

$$SNR_Q(dB) = 10.\log\frac{P_S}{P_N} = 10.\log\frac{\sum_{n=-\infty}^{\infty} x_S^2[n]}{\sum_{n=-\infty}^{\infty} \xi^2[n]}$$

CAPÍTULO 8

REDES DE COMPUTADORAS

INTRODUCCIÓN:

Definimos "redes de computadoras" a un conjunto de computadoras interconectadas. Se dice que dos computadoras están interconectadas si pueden intercambiar información. Los enlaces pueden ser: cables de cobre, fibras ópticas, microondas, rayos infrarrojos, etc.

Una red de computadoras no requiere homogeneidad de hardware ni de software. Las diferentes máquinas pueden tener diferentes procesadores y sistemas operativos.

Definimos "sistema distribuido" a un conjunto de computadoras independientes que aparece ante sus usuarios como un sistema consistente y único.

Una capa de software que se ejecuta sobre el sistema operativo, denominada middleware**,** es la responsable de implementar este modelo. Un ejemplo de un sistema distribuido es World Wide Web**,** en la cual todo se ve como un documento (una página Web).

Entonces, un sistema distribuido es un sistema de software construido sobre una red. Entonces, la diferencia entre una red y un sistema distribuido está en el software más que en el hardware.

Nota: Para más información acerca de los sistemas distribuidos, vea (Tanenbaum y Van Steen, 2002).

8.1 USOS DE LAS REDES DE COMPUTADORAS

8.1.1 Aplicaciones de negocios

Muchas compañías tienen una cantidad considerable de computadoras. Por ejemplo, una compañía podría tener computadoras separadas para supervisar la producción, controlar inventarios y hacer la nómina. Al principio estas computadoras tal vez hayan trabajado por separado, pero, en algún momento, la administración decidió conectarlas para extraer y correlacionar información acerca de toda la compañía.

Dicho de una manera más general, el asunto aquí es **compartir los recursos** y el objetivo es hacer que todos los programas, el equipo y, en particular, los datos estén disponibles para todos los que se conecten a la red, independientemente de la ubicación física del recurso y del usuario.

Un ejemplo claro y muy difundido es el de un grupo de oficinistas que comparten una impresora. Ninguno de los individuos necesita una impresora privada, y una impresora de alto volumen en red suele ser más barata, rápida y fácil de mantener que varias impresoras individuales.

Sin embargo, compartir información es tal vez más importante que compartir recursos físicos, como impresoras, escáneres y quemadores de CDs. Para las compañías grandes y medianas, así como para muchas pequeñas, la información computarizada es vital. La mayoría de las compañías tiene en línea registros de clientes, inventarios, cuentas por cobrar, estados financieros, información de impuestos, etcétera. Si todas las computadoras de un banco se cayeran, éste no duraría más de cinco minutos. Una moderna planta manufacturera, con una línea de ensamblado controlada por computadora, ni siquiera duraría ese tiempo. Incluso una pequeña agencia de viajes o un despacho jurídico de tres personas ahora dependen en gran medida de las

redes de computadoras para que sus empleados puedan tener acceso de manera instantánea a la información y a los documentos importantes.

En las compañías más pequeñas, es posible que todas las computadoras estén en una sola oficina o en un solo edificio, pero en las más grandes, las computadoras y los empleados pueden estar dispersos en docenas de oficinas y plantas en varios países. No obstante, un vendedor en Nueva York podría requerir algunas veces tener acceso a una base de datos de inventario de productos que se encuentra en Singapur. En otras palabras, el hecho de que un usuario esté a 15,000 km de sus datos no debe ser impedimento para que utilice esos datos como si fueran locales. Esta meta se podría resumir diciendo que es un intento por acabar con la "tiranía de la geografía".

En términos aún más sencillos, es posible imaginar el sistema de información de una compañía como si consistiera en una o más bases de datos y algunos empleados que necesitan acceder a ellas de manera remota. En este modelo, los datos están almacenados en computadoras poderosas que se llaman **servidores.** Con frecuencia, éstos se encuentran alojados en una central y un administrador de sistemas les da mantenimiento. En contraste, los empleados tienen en sus escritorios máquinas más sencillas, llamadas **clientes,** con las que pueden acceder a datos remotos —por ejemplo, para incluirlos en las hojas de cálculo que están elaborando. (Algunas veces nos referiremos a los usuarios de las máquinas como "el cliente", pero debe quedar claro, por el contexto, si el término se refiere a la computadora o a su usuario.) Las máquinas cliente y servidor están conectadas por una red, como se ilustra en la figura 8.1. Observe que hemos representado a la red como un óvalo sencillo, sin detalle alguno. Utilizaremos esta forma cuando nos refiramos a una red en sentido general. Cuando se requieran más detalles, los proporcionaremos.

Este conjunto se conoce como **modelo cliente-servidor.** Se utiliza ampliamente y forma la base en gran medida del uso de redes. Es aplicable cuando el cliente y el servidor están en el mismo edificio (por ejemplo, cuando pertenecen a la misma compañía), pero también cuando están bastante retirados. Por ejemplo, cuando una persona en casa accede a una página Web, se emplea el mismo modelo, en el que el servidor remoto de Web es el servidor y la computadora personal del usuario es el cliente. En la mayoría de los casos, un servidor puede manejar una gran cantidad de clientes.

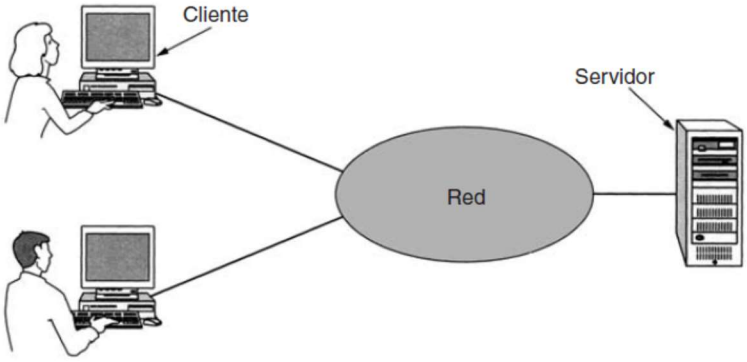


Figura 8.1 – Red con dos clientes y un servidor

Si vemos el modelo cliente-servidor en detalle, nos daremos cuenta de que hay dos procesos involucrados, uno en la máquina cliente y otro en la máquina servidor. La comunicación toma la siguiente forma: el proceso cliente envía una solicitud a través de la red al proceso servidor y espera una respuesta. Cuando el proceso servidor recibe la solicitud, realiza el

trabajo que se le pide o busca los datos solicitados y devuelve una respuesta. Estos mensajes se muestran en la figura 8.2.

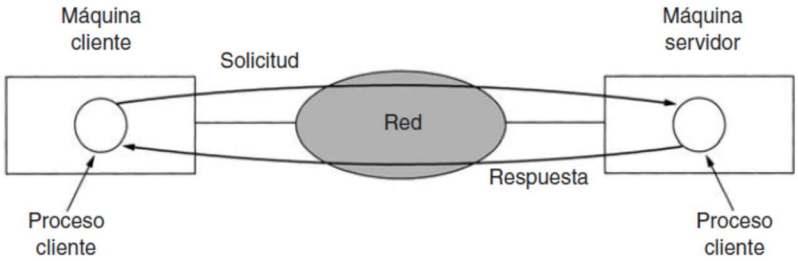


Figura 8.2 - El modelo cliente-servidor implica solicitudes y respuestas.

Un segundo objetivo de la configuración de una red de computadoras tiene que ver más con la gente que con la información e, incluso, con las computadoras mismas. Una red de computadoras es un poderoso **medio de comunicación** entre los empleados. Casi todas las compañías que tienen dos o más computadoras cuentan con **correo electrónico,** mediante el cual los empleados mantienen generalmente una comunicación diaria. De hecho, una queja común es la gran cantidad de correo electrónico que tenemos que atender, mucho de él sin sentido porque los jefes han descubierto que pueden enviar el mismo mensaje (a menudo sin contenido) a todos sus subordinados con sólo oprimir un botón.

Pero el correo electrónico no es la única forma de comunicación mejorada que las redes de computadoras hacen posible. Con una red es fácil que dos o más personas que trabajan a distancia escriban en conjunto un informe. Si un empleado hace un cambio a un documento en línea, los demás pueden ver el cambio de inmediato, en vez de esperar una carta durante varios días. Esta agilización facilita la cooperación entre grupos de personas que no se encuentran en el mismo lugar, lo cual antes había sido imposible.

Otra forma de comunicación asistida por computadora es la videoconferencia. Con esta tecnología, los empleados en ubicaciones distantes pueden tener una reunión, viéndose y escuchándose unos a otros e incluso escribiendo en una pizarra virtual compartida. La videoconferencia es una herramienta poderosa para eliminar el costo y el tiempo que anteriormente se empleaba en viajar. A veces se dice que la comunicación y el transporte están en competencia, y que el que gane hará obsoleto al otro.

Una tercera meta para cada vez más compañías es hacer negocios de manera electrónica con otras compañías, sobre todo proveedores y clientes. Por ejemplo, los fabricantes de automóviles, de aviones, de computadoras, etcétera, compran subsistemas de diversos proveedores y luego ensamblan las partes. Mediante las redes de computadoras los fabricantes pueden hacer pedidos electrónicamente conforme se requieran. Tener la capacidad de hacer pedidos en tiempo real (es decir, conforme se requieren) reduce la necesidad de tener grandes inventarios y mejora la eficiencia.

Una cuarta meta que se está volviendo más importante es la de hacer negocios con consumidores a través de Internet. Las líneas aéreas, las librerías y los vendedores de música han descubierto que muchos consumidores prefieren realizar sus compras desde casa. Por consiguiente, muchas compañías proporcionan en línea catálogos de sus productos y servicios y levantan pedidos de la misma manera. Es lo que se conoce como **comercio electrónico.**

8.1.2 Aplicaciones domésticas

En 1977 Ken Olsen era presidente de Digital Equipment Corporation, que en esa época era el segundo proveedor de computadoras en el mundo (después de IBM). Cuando se le preguntó por qué Digital no perseguía el mercado de las computadoras personales en gran volumen, contestó: "No hay razón alguna para que un individuo tenga una computadora en su casa". La historia demostró lo contrario y Digital ya no existe. ¿Por qué la gente compra computadoras para uso doméstico?

En principio, para procesamiento de texto y juegos, pero en los últimos años esto ha cambiado radicalmente. Tal vez la razón más importante ahora sea por el acceso a Internet. Algunos de los usos más comunes de Internet por parte de usuarios domésticos son los siguientes:

1. Acceso a información remota.
2. Comunicación de persona a persona.
3. Entretenimiento interactivo.
4. Comercio electrónico.

El acceso a la información remota se puede realizar por diversas razones. Puede ser que navegue por World Wide Web para obtener información o sólo por diversión. La información disponible incluye artes, negocios, cocina, gobiernos, salud, historia, pasatiempos, recreación, ciencia, deportes, viajes y muchas otras cosas más. La diversión viene en demasiadas formas como para mencionarlas, más algunas otras que es mejor no mencionar.

Muchos periódicos ahora están disponibles en línea y pueden personalizarse. Por ejemplo, en algunos casos le puede indicar a un periódico que desea toda la información acerca de políticos corruptos, incendios, escándalos que involucran a las celebridades y epidemias, pero nada sobre fútbol. Incluso puede hacer que los artículos que usted desea se descarguen en su disco duro o se impriman mientras usted duerme, para que cuando se levante a desayunar los tenga disponibles.

Mientras continúe esta tendencia, se provocará el desempleo masivo de los niños de 12 años que entregan los diarios, pero los periódicos lo quieren así porque la distribución siempre ha sido el punto débil en la gran cadena de producción.

El tema más importante después de los periódicos (además de las revistas y periódicos científicos) son las bibliotecas digitales en línea. Muchas organizaciones profesionales, como la ACM (*www.acm.org*) y la Sociedad de Computación del IEEE (*www.computer.org*), ya cuentan con muchos periódicos y presentaciones de conferencias en línea. Otros grupos están creciendo de manera rápida. Dependiendo del costo, tamaño y peso de las computadoras portátiles, los libros impresos podrían llegar a ser obsoletos. Los escépticos deben tomar en cuenta el efecto que la imprenta tuvo sobre los manuscritos ilustrados del medioevo.

Todas las aplicaciones anteriores implican las interacciones entre una persona y una base de datos remota llena de información. La segunda gran categoría del uso de redes es la comunicación de persona a persona, básicamente la respuesta del siglo XXI al teléfono del siglo XIX. Millones de personas en todo el mundo utilizan a diario el correo electrónico y su uso está creciendo rápidamente. Ya es muy común que contenga audio y vídeo, así como texto y figuras. Los aromas podrían tardar un poco más.

Muchas personas utilizan los **mensajes instantáneos.** Esta característica, derivada del programa *talk* de UNIX, que se utiliza aproximadamente desde 1970, permite que las personas se escriban mensajes en tiempo real. Una versión, para varias personas, de esta idea es el **salón de**

conversación (*chat room*), en el que un grupo de personas puede escribir mensajes para que todos los vean.

Los grupos de noticias mundiales, con debates sobre todo tema imaginable, ya son un lugar común entre un grupo selecto de personas y este fenómeno crecerá para abarcar a la población en general. Estos debates, en los que una persona envía un mensaje y todos los demás suscriptores del grupo de noticias lo pueden leer, van desde los humorísticos hasta los apasionados. A diferencia de los salones de conversación, los grupos de noticias no son en tiempo real y los mensajes se guardan para que cuando alguien vuelva de vacaciones, encuentre todos los mensajes que hayan sido enviados en el ínterin, esperando pacientemente a ser leídos.

Otro tipo de comunicación de persona a persona a menudo se conoce como comunicación de **igual a igual** (*peer to peer*), para distinguirla del modelo cliente-servidor (Parameswaran y cols., 2001). De esta forma, los individuos que forman un grupo esparcido se pueden comunicar con otros del grupo, como se muestra en la figura 8.3. Cada persona puede, en principio, comunicarse con una o más personas; no hay una división fija de clientes y servidores.

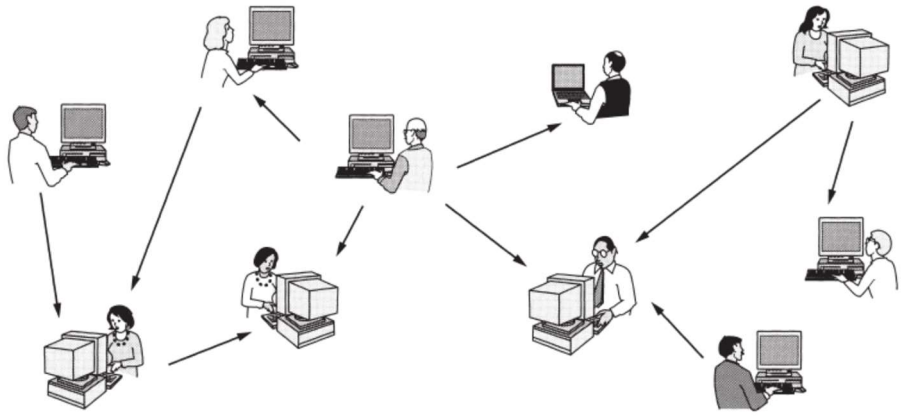


Figura 8.3 - En el sistema de igual a igual no hay clientes ni servidores fijos.

8.1.3 Usuarios móviles

Las computadoras portátiles, como las *notebook* y los asistentes personales digitales (PDAs), son uno de los segmentos de crecimiento más rápido de la industria de la computación.

Muchos propietarios de estas computadoras poseen máquinas de escritorio en la oficina y desean estar conectados a su base doméstica cuando están de viaje o fuera de casa. Puesto que no es posible tener una conexión alámbrica en autos y aviones, hay un gran interés en las redes inalámbricas.

En esta sección veremos brevemente algunos usos de ellas.

¿Por qué querría alguien una? Un argumento común es la oficina portátil. Con frecuencia, las personas que están de viaje desean utilizar sus equipos portátiles para enviar y recibir llamadas telefónicas, faxes y correo electrónico, navegar en Web, acceder a archivos remotos e iniciar sesión en máquinas remotas. Y desean hacer esto desde cualquier punto, ya sea por tierra, mar o aire. Por ejemplo, actualmente en las conferencias por computadora, los organizadores suelen configurar una red inalámbrica en el área de la conferencia. Cualquiera que tenga una computadora portátil y un módem inalámbrico puede conectarse a Internet, como si la computadora estuviera conectada a una red cableada. Del mismo modo, algunas universidades han instalado redes inalámbricas en sus campus para que los estudiantes se puedan sentar entre los árboles y consultar los archivos de la biblioteca o leer su correo electrónico.

Las redes inalámbricas son de gran utilidad para las flotas de camiones, taxis, vehículos de entrega y reparadores, para mantenerse en contacto con la casa. Por ejemplo, en muchas ciudades los taxistas trabajan por su cuenta, más que para una empresa de taxis. En algunas de estas ciudades, los taxis tienen una pantalla que el conductor puede ver. Cuando el cliente solicita un servicio, un despachador central escribe los puntos en los que el chofer deberá recoger y dejar al cliente. Esta información se despliega en las pantallas de los conductores y suena un timbre. El conductor que oprima primero un botón en la pantalla recibe la llamada.

Las redes inalámbricas también son importantes para la milicia. Si tiene que estar disponible en breve para pelear una guerra en cualquier parte de la Tierra, probablemente no sea bueno pensar en utilizar la infraestructura de conectividad de redes local. Lo mejor sería tener la propia.

Aunque la conectividad inalámbrica y la computación portátil se relacionan frecuentemente, no son idénticas, como se muestra en la figura 8.4, en la que vemos una diferencia entre **inalámbrica fija** e **inalámbrica móvil.** Incluso en ocasiones las computadoras portátiles son cableadas. Por ejemplo, si un viajero conecta una portátil a una toma telefónica en su habitación del hotel, tiene movilidad sin una red inalámbrica.

Inalámbrica	Móvil	Aplicaciones
No	No	Computadoras de escritorio en oficinas
No	Sí	Una computadora portátil usada en un cuarto de hotel
Sí	No	Redes en construcciones antiguas sin cableado
Sí	Sí	Oficina portátil; PDA para inventario de almacén

Figura 8.4 - Combinaciones de redes inalámbricas y computación móvil.

8.1.4 Temas sociales

La amplia introducción de las redes ha presentado problemas sociales, éticos y políticos. Mencionemos brevemente algunos de ellos; un estudio completo requeriría todo un libro, por lo menos.

Un rasgo popular de muchas redes son los **grupos de noticias** o boletines electrónicos mediante los cuales las personas pueden intercambiar mensajes con individuos de los mismos intereses. Siempre y cuando los asuntos se restrinjan a temas técnicos o pasatiempos como la jardinería, no surgirán demasiados problemas. El problema viene cuando los grupos de noticias se enfocan en temas que las personas en realidad tocan con cuidado, como política, religión o sexo. Los puntos de vista enviados a tales grupos podrían ser ofensivos para algunas personas. Peor aún, podrían no ser políticamente correctos.

Además, los mensajes no tienen que limitarse a texto. En la actualidad se pueden enviar fotografías en alta resolución e incluso pequeños videoclips a través de redes de computadoras.

Algunas personas practican la filosofía de vive y deja vivir, pero otras sienten que enviar cierto material (por ejemplo, ataques a países o religiones en particular, pornografía, etcétera) es sencillamente inaceptable y debe ser censurado. Los diversos países tienen diferentes y conflictivas leyes al respecto. De esta manera, el debate se aviva.

Las personas han demandado a los operadores de redes, afirmando que son responsables, como sucede en el caso de los periódicos y las revistas, del contenido que transmiten. La respuesta inevitable es que una red es como una compañía de teléfonos o la oficina de correos, por lo que no se puede esperar que vigilen lo que dicen los usuarios. Más aún, si los operadores de redes censuraran los mensajes, borrarían cualquier contenido que contuviera incluso la

mínima posibilidad de que se les demandara, pero con esto violarían los derechos de sus usuarios a la libre expresión. Probablemente lo más seguro sería decir que este debate seguirá durante algún tiempo.

Otra área divertida es la de los derechos de los empleados en comparación con los de los empleadores. Muchas personas leen y escriben correo electrónico en el trabajo. Muchos empleadores han exigido el derecho a leer y, posiblemente, censurar los mensajes de los empleados, incluso los enviados desde un equipo doméstico después de las horas de trabajo. No todos los empleados están de acuerdo con esto.

Incluso si los empleadores tienen poder sobre los empleados, ¿esta relación también rige a las universidades y los estudiantes? ¿Qué hay acerca de las escuelas secundarias y los estudiantes? En 1994, la Carnegie-Mellon University decidió suspender el flujo de mensajes entrantes de varios grupos de noticias que trataban sexo porque la universidad sintió que el material era inapropiado para menores (es decir, menores de 18 años). Tomó años recuperarse de este suceso.

Otro tema de importancia es el de los derechos del gobierno y los de los ciudadanos. El FBI ha instalado un sistema en muchos proveedores de servicios de Internet para curiosear entre todos los correos electrónicos en busca de fragmentos que le interesen (Blaze y Bellovin, 2000; Sobel, 2001; Zacks, 2001). El sistema se llamaba originalmente **Carnivore** pero la mala publicidad provocó que se cambiara el nombre por uno menos agresivo que sonara como DCS1000. Pero su objetivo sigue siendo el de espiar a millones de personas con la esperanza de encontrar información acerca de actividades ilegales. Por desgracia, la Cuarta Enmienda de la Constitución de Estados Unidos prohíbe que el gobierno realice investigaciones sin una orden de cateo. Decidir si estas palabras, escritas en el siglo XVIII, aún son válidas en el siglo XXI es un asunto que podría mantener ocupadas a las cortes hasta el siglo XXII.

8.2 HARDWARE DE REDES

Ya es tiempo de centrar nuevamente la atención en los temas técnicos correspondientes al diseño de redes (la parte de trabajo) y dejar a un lado las aplicaciones y los aspectos sociales de la conectividad (la parte divertida). Por lo general, no hay una sola clasificación aceptada en la que se ajusten todas las redes de computadoras, pero hay dos que destacan de manera importante: la tecnología de transmisión y la escala. Examinaremos cada una a la vez.

En un sentido amplio, hay dos tipos de tecnología de transmisión que se utilizan de manera extensa. Son las siguientes:

1. Enlaces de difusión.

2. Enlaces de punto a punto.

Las **redes de difusión** (*broadcast*) tienen un solo canal de comunicación, por lo que todas las máquinas de la red lo comparten. Si una máquina envía un mensaje corto —en ciertos contextos conocido como **paquete**—, todas las demás lo reciben. Un campo de dirección dentro del paquete especifica el destinatario. Cuando una máquina recibe un paquete, verifica el campo de dirección.

Si el paquete va destinado a esa máquina, ésta lo procesa; si va destinado a alguna otra, lo ignora. En una analogía, imagine a alguien que está parado al final de un corredor con varios cuartos a los lados y que grita: “Jorge, ven. Te necesito”. Aunque en realidad el grito (paquete) podría haber sido escuchado (recibido), por muchas personas, sólo Jorge responde (lo procesa). Los demás simplemente lo ignoran. Otra analogía es la de los anuncios en un aeropuerto que piden a todos los pasajeros del vuelo 644 se reporten en la puerta 12 para abordar de inmediato.

Por lo general, los sistemas de difusión también permiten el direccionamiento de un paquete a *todos* los destinos utilizando un código especial en el campo de dirección. Cuando se transmite un paquete con este código, todas las máquinas de la red lo reciben y procesan. Este modo de operación se conoce como **difusión** (*broadcasting*). Algunos sistemas de difusión también soportan la transmisión a un subconjunto de máquinas, algo conocido como **multidifusión** (*multicasting*).

Un esquema posible es la reserva de un bit para indicar la multidifusión. Los bits de dirección $n - 1$ restantes pueden contener un número de grupo. Cada máquina puede "suscribirse" a alguno o a todos los grupos. Cuando se envía un paquete a cierto grupo, se distribuye a todas las máquinas que se suscriben a ese grupo.

En contraste, las redes **punto a punto** constan de muchas conexiones entre pares individuales de máquinas. Para ir del origen al destino, un paquete en este tipo de red podría tener que visitar primero una o más máquinas intermedias. A menudo es posible que haya varias rutas o longitudes diferentes, de manera que encontrar las correctas es importante en redes de punto a punto. Por regla general (aunque hay muchas excepciones), las redes más pequeñas localizadas en una misma área geográfica tienden a utilizar la difusión, mientras que las más grandes suelen ser de punto a punto. La transmisión de punto a punto con un emisor y un receptor se conoce como **unidifusión** (*unicasting*).

Un criterio alternativo para la clasificación de las redes es su escala. En la figura 8.5 clasificamos los sistemas de procesadores múltiples por tamaño físico. En la parte superior se muestran las **redes de área personal,** que están destinadas para una sola persona. Por ejemplo, una red inalámbrica que conecta una computadora con su ratón, teclado e impresora, es una red de área personal.

Incluso un PDA que controla el audífono o el marcapaso de un usuario encaja en esta categoría. A continuación de las redes de área personal se encuentran redes más grandes. Se pueden dividir en redes de área local, de área metropolitana y de área amplia. Por último, la conexión de dos o más redes se conoce como interred.

Distancia entre procesadores	Procesadores ubicados en el mismo	Ejemplo
1 m	Metro cuadrado	Red de área personal
10 m	Cuarto	Red de área local
100 m	Edificio	
1 km	Campus	
10 km	Ciudad	Red de área metropolitana
100 km	País	Red de área amplia
1,000 km	Continente	
10,000 km	Planeta	Internet

Figura 8.5 - Clasificación de procesadores interconectados por escala.

8.2.1 Redes de área local

Las **redes de área local** (generalmente conocidas como **LANs**) son redes de propiedad privada que se encuentran en un solo edificio o en un campus de pocos kilómetros de longitud. Se utilizan ampliamente para conectar computadoras personales y estaciones de trabajo en oficinas de una empresa y de fábricas para compartir recursos (por ejemplo, impresoras) e intercambiar información.

Las LANs son diferentes de otros tipos de redes en tres aspectos: 1) tamaño; 2) tecnología de transmisión y 3) topología.

Las LANs están restringidas por tamaño, es decir, el tiempo de transmisión en el peor de los casos es limitado y conocido de antemano. El hecho de conocer este límite permite utilizar ciertos tipos de diseño, lo cual no sería posible de otra manera. Esto también simplifica la administración de la red.

Las LANs podrían utilizar una tecnología de transmisión que consiste en un cable al cual están unidas todas las máquinas, como alguna vez lo estuvo parte de las líneas de las compañías telefónicas en áreas rurales. Las LANs tradicionales se ejecutan a una velocidad de 10 a 100 Mbps, tienen un retardo bajo (microsegundos o nanosegundos) y cometen muy pocos errores. Las LANs más nuevas funcionan hasta a 10 Gbps. Nosotros continuaremos con lo tradicional y mediremos las velocidades de las líneas en megabits por segundo (1 Mbps es igual a 1,000,000 de bits por segundo) y gigabits por segundo (1 Gbps es igual a 1,000,000,000 de bits por segundo). Para las LANs de difusión son posibles varias topologías. La figura 8.6 muestra dos de ellas.

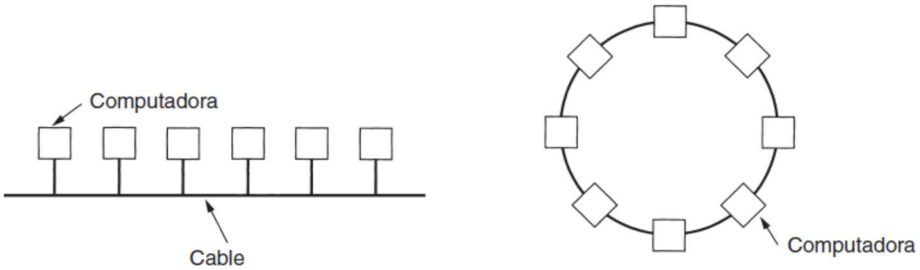


Figura 8.6 - Dos redes de difusión. (a) De bus. (b) De anillo.

En una red de bus (es decir, un cable lineal), en cualquier instante al menos una máquina es la maestra y puede transmitir. Todas las demás máquinas se abstienen de enviar. Cuando se presenta el conflicto de que dos o más máquinas desean transmitir al mismo tiempo, se requiere un mecanismo de arbitraje. Tal mecanismo podría ser centralizado o distribuido. Por ejemplo, el IEEE 802.3, popularmente conocido como **Ethernet,** es una red de difusión basada en bus con control descentralizado, que por lo general funciona de 10 Mbps a 10 Gbps. Las computadoras que están en una Ethernet pueden transmitir siempre que lo deseen; si dos o más paquetes entran en colisión, cada computadora espera un tiempo aleatorio y lo intenta de nuevo más tarde.

8.2.2 Redes de área metropolitana

Una **red de área metropolitana** (**MAN**) abarca una ciudad. El ejemplo más conocido de una MAN es la red de televisión por cable disponible en muchas ciudades. Este sistema creció a partir de los primeros sistemas de antena comunitaria en áreas donde la recepción de la televisión al aire era pobre. En dichos sistemas se colocaba una antena grande en la cima de una colina cercana y la señal se canalizaba a las casas de los suscriptores.

Al principio eran sistemas diseñados de manera local con fines específicos. Después las compañías empezaron a pasar a los negocios, y obtuvieron contratos de los gobiernos de las ciudades para cablear toda una ciudad. El siguiente paso fue la programación de televisión e incluso canales designados únicamente para cable. Con frecuencia, éstos emitían programas de un solo tema, como sólo noticias, deportes, cocina, jardinería, etcétera. Sin embargo, desde su inicio y hasta finales de la década de 1990, estaban diseñados únicamente para la recepción de televisión.

A partir de que Internet atrajo una audiencia masiva, los operadores de la red de TV por cable se dieron cuenta de que, con algunos cambios al sistema, podrían proporcionar servicio de Internet de dos vías en las partes sin uso del espectro. En ese punto, el sistema de TV por cable empezaba a transformarse de una forma de distribución de televisión a una red de área metropolitana. Para que se dé una idea, una MAN podría verse como el sistema que se muestra en la figura 8.7, donde se aprecia que las señales de TV e Internet se alimentan hacia un **amplificador head end** para enseguida transmitirse a las casas de las personas.

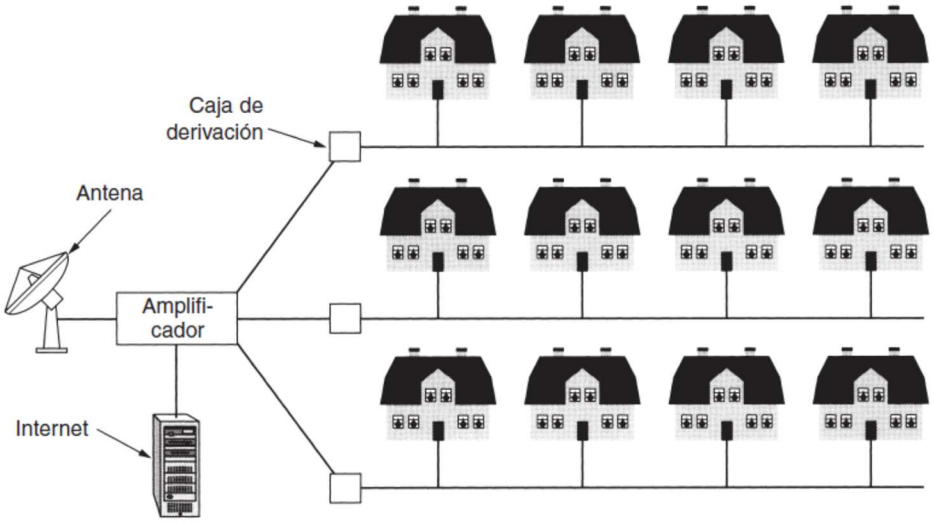


Figura 8.7 - Una red de área metropolitana, basada en TV por cable.

8.2.3 Redes de área amplia

Una red de área amplia (WAN), abarca una gran área geográfica, con frecuencia un país o un continente. Contiene un conjunto de máquinas diseñado para programas (es decir, aplicaciones) de usuario. Seguiremos el uso tradicional y llamaremos hosts a estas máquinas. Los hosts están conectados por una subred de comunicación, o simplemente subred, para abreviar. Los clientes son quienes poseen a los hosts (es decir, las computadoras personales de los usuarios), mientras que, por lo general, las compañías telefónicas o los proveedores de servicios de Internet poseen y operan la subred de comunicación. La función de una subred es llevar mensajes de un *host* a otro, como lo hace el sistema telefónico con las palabras del que habla al que escucha. La separación de los aspectos de la comunicación pura de la red (la subred) de los aspectos de la aplicación (los *hosts*), simplifica en gran medida todo el diseño de la red.

En la mayoría de las redes de área amplia la subred consta de dos componentes distintos: líneas de transmisión y elementos de conmutación. Las **líneas de transmisión** mueven bits entre máquinas.

Pueden estar hechas de cable de cobre, fibra óptica o, incluso, radioenlaces. Los **elementos de conmutación** son computadoras especializadas que conectan tres o más líneas de transmisión.

Cuando los datos llegan a una línea de entrada, el elemento de conmutación debe elegir una línea de salida en la cual reenviarlos. Estas computadoras de conmutación reciben varios nombres; conmutadores y enrutadores son los más comunes.

En este modelo, que se muestra en la figura 8.8, cada *host* está conectado frecuentemente a una LAN en la que existe un enrutador, aunque en algunos casos un *host* puede estar conectado

de manera directa a un enrutador. El conjunto de líneas de comunicación y enrutadores (pero no de *hosts*) forma la subred.

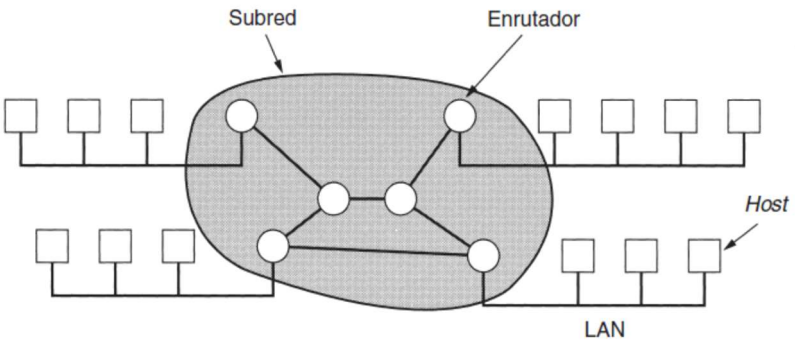


Figura 8.8 - Relación entre *hosts* de LANs y la subred.

A continuación, se presenta un breve comentario acerca del término "subred". Originalmente, su **único** significado era el conjunto de enrutadores y líneas de comunicación que movía paquetes del *host* de origen al de destino. Sin embargo, algunos años más tarde también adquirió un segundo significado junto con el direccionamiento de redes. Desgraciadamente, no existe una alternativa de amplio uso con respecto a su significado inicial por lo que, con algunas reservas, utilizaremos este término en ambos sentidos. El contexto dejará en claro su significado.

En la mayoría de las WANs, la red contiene numerosas líneas de transmisión, cada una de las cuales conecta un par de enrutadores. Si dos enrutadores que no comparten una línea de transmisión quieren conectarse, deberán hacerlo de manera indirecta, a través de otros enrutadores. Cuando un paquete es enviado desde un enrutador a otro a través de uno o más enrutadores intermedios, el paquete se recibe en cada enrutador intermedio en su totalidad, se almacena ahí hasta que la línea de salida requerida esté libre y, por último, se reenvía. Una subred organizada a partir de este principio se conoce como subred de **almacenamiento y reenvío** (*store and forward*) o de **conmutación de paquetes.** Casi todas las redes de área amplia (excepto las que utilizan satélites) tienen subredes de almacenamiento y reenvío. Cuando los paquetes son pequeños y tienen el mismo tamaño, se les llama **celdas.**

El principio de una WAN de conmutación de paquetes es tan importante que vale la pena dedicarle algunas palabras más. En general, cuando un proceso de cualquier *host* tiene un mensaje que se va a enviar a un proceso de algún otro *host*, el *host* emisor divide primero el mensaje en paquetes, los cuales tienen un número de secuencia. Estos paquetes se envían entonces por la red de uno en uno en una rápida sucesión. Los paquetes se transportan de forma individual a través de la red y se depositan en el *host* receptor, donde se reensamblan en el mensaje original y se entregan al proceso receptor. En la figura 8.9 se ilustra un flujo de paquetes correspondiente a algún mensaje inicial.

En esta figura todos los paquetes siguen la ruta *ACE* en vez de la *ABDE* o *ACDE*. En algunas redes todos los paquetes de un mensaje determinado *deben* seguir la misma ruta; en otras, cada paquete se enruta por separado. Desde luego, si *ACE* es la mejor ruta, todos los paquetes se podrían enviar a través de ella, incluso si cada paquete se enruta de manera individual.

Las decisiones de enrutamiento se hacen de manera local. Cuando un paquete llega al enrutador *A*, éste debe decidir si el paquete se enviará hacia *B* o hacia *C*. La manera en que el enrutador *A* toma esa decisión se conoce como **algoritmo de enrutamiento.**

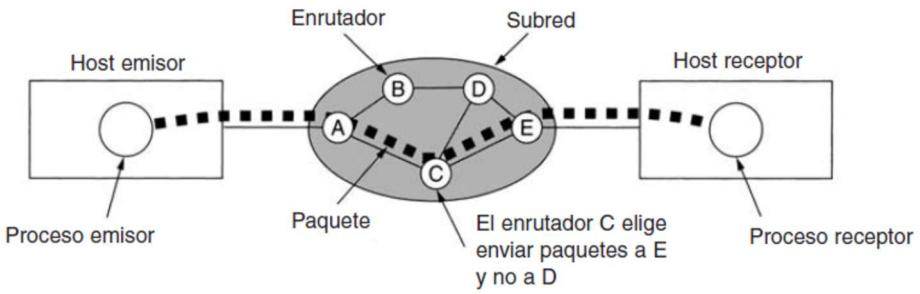


Figura 8.9 - Flujo de paquetes desde un emisor a un receptor.

8.2.4 Redes inalámbricas

La comunicación inalámbrica digital no es una idea nueva. A principios de 1901, el físico italiano Guillermo Marconi demostró un telégrafo inalámbrico desde un barco a tierra utilizando el código Morse (después de todo, los puntos y rayas son binarios). Los sistemas inalámbricos digitales de la actualidad tienen un mejor desempeño, pero la idea básica es la misma.

Como primera aproximación, las redes inalámbricas se pueden dividir en tres categorías principales:

1. Interconexión de sistemas.
2. LANs inalámbricas.
3. WANs inalámbricas.

La interconexión de sistemas se refiere a la interconexión de componentes de una computadora que utiliza radio de corto alcance. La mayoría de las computadoras tiene un monitor, teclado, ratón e impresora, conectados por cables a la unidad central. Son tantos los usuarios nuevos que tienen dificultades para conectar todos los cables en los enchufes correctos (aun cuando suelen estar codificados por colores) que la mayoría de los proveedores de computadoras ofrece la opción de enviar a un técnico a la casa del usuario para que realice esta tarea. En consecuencia, algunas compañías se reunieron para diseñar una red inalámbrica de corto alcance llamada **Bluetooth** para conectar sin cables estos componentes. Bluetooth también permite conectar cámaras digitales, auriculares, escáneres y otros dispositivos a una computadora con el único requisito de que se encuentren dentro del alcance de la red. Sin cables, sin instalación de controladores, simplemente se colocan, se encienden y funcionan. Para muchas personas, esta facilidad de operación es algo grandioso.

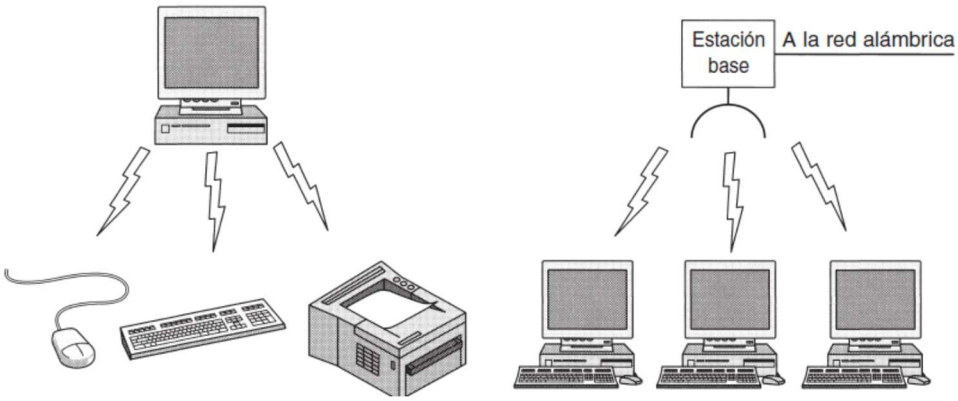


Figura 8.10 - (a) Configuración Bluetooth. (b) LAN inalámbrica.

En la forma más sencilla, las redes de interconexión de sistemas utilizan el paradigma del maestro y el esclavo de la figura 8.10 (a). La unidad del sistema es, por lo general, el maestro que trata al ratón, al teclado, etcétera, como a esclavos. El maestro le dice a los esclavos qué direcciones utilizar, cuándo pueden difundir, durante cuánto tiempo pueden transmitir, qué frecuencias pueden utilizar, etcétera.

El siguiente paso en la conectividad inalámbrica son las LANs inalámbricas. Son sistemas en los que cada computadora tiene un módem de radio y una antena mediante los que se puede comunicar con otros sistemas. En ocasiones, en el techo se coloca una antena con la que las máquinas se comunican, como se ilustra en la figura 8.10 (b). Sin embargo, si los sistemas están lo suficientemente cerca, se pueden comunicar de manera directa entre sí en una configuración de igual a igual. Las LANs inalámbricas se están haciendo cada vez más comunes en casas y oficinas pequeñas, donde instalar Ethernet se considera muy problemático, así como en oficinas ubicadas en edificios antiguos, cafeterías de empresas, salas de conferencias y otros lugares. Existe un estándar para las LANs inalámbricas, llamado **IEEE 802.11,** que la mayoría de los sistemas implementa y que se ha extendido ampliamente.

El tercer tipo de red inalámbrica se utiliza en sistemas de área amplia. La red de radio utilizada para teléfonos celulares es un ejemplo de un sistema inalámbrico de banda ancha baja. Este sistema ha pasado por tres generaciones. La primera era analógica y sólo para voz. La segunda era digital y sólo para voz. La tercera generación es digital y es tanto para voz como para datos. En cierto sentido, las redes inalámbricas celulares son como las LANs inalámbricas, excepto porque las distancias implicadas son mucho más grandes y las tasas de bits son mucho más bajas. Las LANs inalámbricas pueden funcionar a tasas de hasta 50 Mbps en distancias de decenas de metros.

Los sistemas celulares funcionan debajo de 1 Mbps, pero la distancia entre la estación base y la computadora o teléfono se mide en kilómetros más que en metros. Además de estas redes de baja velocidad, también se han desarrollado las redes inalámbricas de área amplia con alto ancho de banda. El enfoque inicial es el acceso inalámbrico a Internet a alta velocidad, desde los hogares y las empresas, dejando a un lado el sistema telefónico. Este servicio se suele llamar servicio de distribución local multipuntos. También se ha desarrollado un estándar para éste, llamado IEEE 802.16.

La mayoría de las redes inalámbricas se enlaza a la red alámbrica en algún punto para proporcionar acceso a archivos, bases de datos e Internet. Hay muchas maneras de efectuar estas conexiones, dependiendo de las circunstancias.

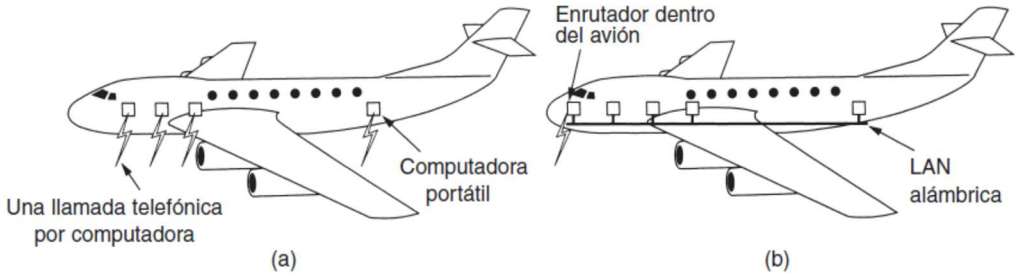


Figura 8.11 - (a) Computadoras móviles individuales. (b) LAN dentro del avión.

Por ejemplo, en la figura 8.11 (a) mostramos un aeroplano con una serie de personas que utilizan módems y los teléfonos de los respaldos para llamar a la oficina. Cada llamada es independiente de las demás. Sin embargo, una opción mucho más eficiente es la LAN dentro del

avión de la figura 8.11 (b), donde cada asiento está equipado con un conector Ethernet al cual los pasajeros pueden acoplar sus computadoras. El avión tiene un solo enrutador, el cual mantiene un enlace de radio con algún enrutador que se encuentre en tierra, y cambia de enrutador conforme avanza el vuelo. Esta configuración es una LAN tradicional, excepto porque su conexión al mundo exterior se da mediante un enlace por radio en lugar de una línea cableada.

8.2.5 Redes domésticas

La conectividad doméstica está en el horizonte. La idea fundamental es que en el futuro la mayoría de los hogares estarán preparados para conectividad de redes. Cualquier dispositivo del hogar será capaz de comunicarse con todos los demás dispositivos y todos podrán accederse por Internet. Éste es uno de esos conceptos visionarios que nadie solicitó (como los controles remotos de TV o los teléfonos celulares), pero una vez que han llegado nadie se puede imaginar cómo habían podido vivir sin ellos.

Muchos dispositivos son capaces de estar conectados en red. Algunas de las categorías más evidentes (con ejemplos) son las siguientes:

1. Computadoras (de escritorio, portátiles, PDAs, periféricos compartidos).

2. Entretenimiento (TV, DVD, videocámara, cámara fotográfica, estereofónicos, MP3).

3. Telecomunicaciones (teléfono, teléfono móvil, intercomunicadores, fax).

4. Aparatos electrodomésticos (horno de microondas, refrigerador, reloj, horno, aire acondicionado, luces).

5. Telemetría y vigilancia (utilidades para medición, alarma contra fuego y robo, termostato, cámaras inalámbricas).

La conectividad de computadoras domésticas ya está aquí. Muchas casas ya cuentan con un dispositivo para conectar varias computadoras para una conexión rápida a Internet.

El entretenimiento por red ya existe, en cuanto más y más música y películas se puedan descargar de Internet, habrá más demanda para que los equipos de audio y las televisiones que se conectan a Internet. Incluso las personas que desean compartir sus propios vídeos con amigos y familiares, usan una conexión en ambos sentidos. Los dispositivos de telecomunicaciones no sólo están conectados al mundo exterior de forma analógica: ya se comercializan TV digitales con capacidad de funcionar sobre Internet. También existen cajas que adaptan un TV analógico a Internet y la red del hogar. Por último, el monitoreo remoto de la casa y su contenido es el probable ganador. Es muy factible que muchos padres deseen invertir en monitorear con sus PDAs a sus bebés dormidos cuando van a cenar fuera de casa, aun cuando contraten a una niñera. Si bien podemos imaginar una red separada para cada área de aplicación, la integración de todas en una sola red es probablemente una mejor idea.

La conectividad doméstica tiene algunas propiedades diferentes a las de otro tipo de redes. Primero, la red y los dispositivos deben ser fáciles de instalar. El autor ha instalado numerosas piezas de hardware y software en varias computadoras durante varios años con resultados diferentes. Al realizar una serie de llamadas telefónicas al personal de soporte técnico del proveedor por lo general recibió respuestas como: 1) Lea el manual; 2) Reinicie la computadora; 3) Elimine todo el hardware y software, excepto los nuestros, y pruebe de nuevo; 4) Descargue de nuestro sitio Web el controlador más reciente y, si todo eso falla, 5) Reformatee el disco duro y reinstale Windows desde el DVD. Decirle al comprador de un refrigerador con capacidad de Internet que descargue e instale una nueva versión del sistema operativo del refrigerador, no conduce a tener clientes contentos. Los usuarios de computadoras están acostumbrados a soportar productos que no funcionan; los clientes que compran automóviles, televisiones y

refrigeradores son mucho menos tolerantes y esperan productos que trabajen al 100% desde que se compran.

Segundo, la red y los dispositivos deben estar plenamente probados en operación. Los equipos de aire acondicionado solían tener una perilla con cuatro parámetros: OFF, LOW, MEDIUM y HIGH (apagado, bajo, medio, alto). Ahora tienen manuales de 30 páginas. Una vez que puedan conectarse en red, no se le haga extraño que tan sólo el capítulo de seguridad tenga 30 páginas. Esto estará más allá de la comprensión de prácticamente todos los usuarios.

Tercero, el precio bajo es esencial para el éxito. Muy pocas personas, si no es que ninguna, pagarán un precio adicional de $50 por un termostato con capacidad de Internet, debido a que no considerarán que monitorear la temperatura de sus casas desde sus trabajos sea algo importante. Tal vez por $5 sí lo comprarían.

Cuarto, la principal aplicación podría implicar multimedia, por lo que la red necesita capacidad suficiente. No hay mercado para televisiones conectadas a Internet que proyecten películas inseguras a una resolución de 320 · 240 píxeles y 10 cuadros por segundo. Fast Ethernet, el caballo de batalla en la mayoría de las oficinas, no es bastante buena para multimedia. En consecuencia, para que las redes domésticas lleguen a ser productos masivos en el mercado, requerirán mejor desempeño que el de las redes de oficina actuales, así como precios más bajos.

Quinto, se podría empezar con uno o dos dispositivos y expandir de manera gradual el alcance de la red. Esto significa que no habrá problemas con el formato. Decir a los consumidores que adquieran periféricos con interfaces IEEE 1394 (FireWire) y años después retractarse y decir que USB 3.0 es la interfaz del futuro, es hacer clientes caprichosos. La interfaz de red tendrá que permanecer estable durante algunos años.

Sexto, la seguridad y la confianza serán muy importantes. Perder algunos archivos por un virus de correo electrónico es una cosa; que un ladrón desarme su sistema de seguridad desde su PDA y luego saquee su casa es algo muy diferente.

Una pregunta interesante es si las redes domésticas serán cableadas o inalámbricas. La mayoría de los hogares ya tiene seis redes instaladas: electricidad, teléfono, televisión por cable, agua, gas y alcantarillado. Agregar una séptima durante la construcción de una casa no es difícil, pero acondicionar las casas existentes para agregar dicha red es costoso. Los costos favorecen la conectividad inalámbrica, pero la seguridad favorece la conectividad alámbrica. El problema con la conectividad inalámbrica es que las ondas de radio que utiliza traspasan las paredes con mucha facilidad. No a todos les gusta la idea de que cuando vaya a imprimir, se tope con la conexión de su vecino y pueda leer el correo electrónico de éste. En el capítulo 7 estudiamos cómo se puede utilizar la encriptación para proporcionar seguridad, pero en el contexto de una red doméstica la seguridad debe estar bien probada, incluso para usuarios inexpertos. Es más fácil decirlo que hacerlo, incluso en el caso de usuarios expertos.

Para abreviar, la conectividad doméstica ofrece muchas oportunidades y retos. La mayoría de ellos se relaciona con la necesidad de que sean fáciles de manejar, confiables y seguros, en particular en manos de usuarios no técnicos, y que al mismo tiempo proporcionen alto desempeño a bajo costo.

8.2.6 Interredes

Existen muchas redes en el mundo, a veces con hardware y software diferentes. Con frecuencia, las personas conectadas a una red desean comunicarse con personas conectadas a otra red diferente.

La satisfacción de este deseo requiere que se conecten diferentes redes, con frecuencia incompatibles, a veces mediante máquinas llamadas **puertas de enlace** (*gateways*) para hacer la conexión y proporcionar la traducción necesaria, tanto en términos de hardware como de software.

Un conjunto de redes interconectadas se llama **interred.**

Una forma común de interred es el conjunto de LANs conectadas por una WAN. De hecho, si tuviéramos que reemplazar la etiqueta "subred" en la figura 8.8 por "WAN", no habría nada más que cambiar en la figura. En este caso, la única diferencia técnica real entre una subred y una WAN es si hay *hosts* presentes. Si el sistema que aparece en el área gris contiene solamente enrutadores, es una subred; si contiene enrutadores y *hosts*, es una WAN. Las diferencias reales se relacionan con la propiedad y el uso.

Subredes, redes e interredes con frecuencia se confunden. La subred tiene más sentido en el contexto de una red de área amplia, donde se refiere a un conjunto de enrutadores y líneas de comunicación poseídas por el operador de redes. Como una analogía, el sistema telefónico consta de oficinas de conmutación telefónica que se conectan entre sí mediante líneas de alta velocidad, y a los hogares y negocios, mediante líneas de baja velocidad. Estas líneas y equipos, poseídas y administradas por la compañía de teléfonos, forman la subred del sistema telefónico. Los teléfonos mismos (los *hosts* en esta analogía) no son parte de la subred. La combinación de una subred y sus *hosts* forma una red. En el caso de una LAN, el cable y los *hosts* forman la red. En realidad, ahí no hay una subred.

Una interred se forma cuando se interconectan redes diferentes. Desde nuestro punto de vista, al conectar una LAN y una WAN o conectar dos LANs se forma una interred, pero existe poco acuerdo en la industria en cuanto a la terminología de esta área. Una regla de oro es que, si varias empresas pagaron por la construcción de diversas partes de la red y cada una mantiene su parte, tenemos una interred más que una sola red. Asimismo, si la terminología subyacente es diferente en partes diferentes (por ejemplo, difusión y punto a punto), probablemente tengamos dos redes.

8.3 SOFTWARE DE REDES

Las primeras redes de computadoras se diseñaron teniendo al hardware como punto principal y al software como secundario. Esta estrategia ya no funciona. Actualmente el software de redes está altamente estructurado. En las siguientes secciones examinaremos en detalle la técnica de estructuración de software. El método descrito aquí es la clave de todo el libro y se presentará con mucha frecuencia más adelante.

8.3.1 Jerarquías de protocolos

Para reducir la complejidad de su diseño, la mayoría de las redes está organizada como una pila de **capas** o **niveles,** cada una construida a partir de la que está debajo de ella. El número de capas, así como el nombre, contenido y función de cada una de ellas difieren de red a red. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores, a las cuales no se les muestran los detalles reales de implementación de los servicios ofrecidos.

Este concepto es muy conocido y utilizado en la ciencia computacional, donde se conoce de diversas maneras, como ocultamiento de información, tipos de datos abstractos, encapsulamiento de datos y programación orientada a objetos. La idea básica es que una pieza particular de software (o hardware) proporciona un servicio a sus usuarios, pero nunca les muestra los detalles de su estado interno ni sus algoritmos.

La capa *n* de una máquina mantiene una conversación con la capa *n* de otra máquina. Las reglas y convenciones utilizadas en esta conversación se conocen de manera colectiva como protocolo de capa *n*. Básicamente, un **protocolo** es un acuerdo entre las partes en comunicación sobre cómo se debe llevar a cabo la comunicación. Como una analogía, cuando se presenta una mujer con un hombre, ella podría elegir no darle la mano. Él, a su vez, podría decidir saludarla de mano o de beso, dependiendo, por ejemplo, de si es una abogada americana o una princesa europea en una reunión social formal. Violar el protocolo hará más difícil la comunicación, si no es que imposible.

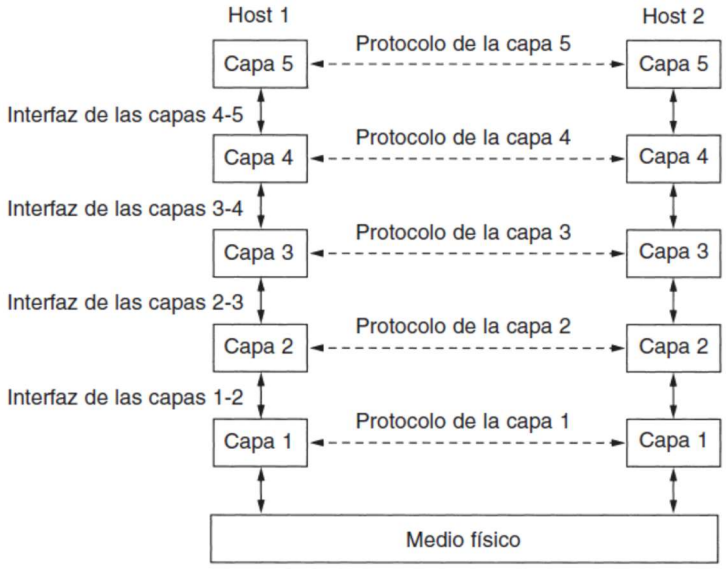


Figura 8.12 - Capas, protocolos e interfaces.

En la figura 8.12 se ilustra una red de cinco capas. Las entidades que abarcan las capas correspondientes en diferentes máquinas se llaman **iguales** (*peers*). Los iguales podrían ser procesos, dispositivos de hardware o incluso seres humanos. En otras palabras, los iguales son los que se comunican a través del protocolo.

En realidad, los datos no se transfieren de manera directa desde la capa *n* de una máquina a la capa *n* de la otra máquina, sino que cada capa pasa los datos y la información de control a la capa inmediatamente inferior, hasta que se alcanza la capa más baja. Debajo de la capa 1 se encuentra el **medio físico** a través del cual ocurre la comunicación real. En la figura 8.12, la comunicación virtual se muestra con líneas punteadas, en tanto que la física, con líneas sólidas.

Entre cada par de capas adyacentes está una **interfaz.** Ésta define qué operaciones y servicios primitivos pone la capa más baja a disposición de la capa superior inmediata. Cuando los diseñadores de redes deciden cuántas capas incluir en una red y qué debe hacer cada una, una de las consideraciones más importantes es definir interfaces limpias entre las capas. Hacerlo así, a su vez, requiere que la capa desempeñe un conjunto específico de funciones bien entendidas.

Además de minimizar la cantidad de información que se debe pasar entre las capas, las interfaces bien definidas simplifican el reemplazo de la implementación de una capa con una implementación totalmente diferente (por ejemplo, todas las líneas telefónicas se reemplazan con canales por satélite) porque todo lo que se pide de la nueva implementación es que ofrezca exactamente el mismo conjunto de servicios a su vecino de arriba, como lo hacía la implementación anterior. De hecho, es muy común que diferentes *hosts* utilicen diferentes implementaciones.

Un conjunto de capas y protocolos se conoce como **arquitectura de red.** La especificación de una arquitectura debe contener información suficiente para permitir que un implementador escriba el programa o construya el hardware para cada capa de modo que se cumpla correctamente con el protocolo apropiado. Ni los detalles de la implementación ni las especificaciones de las interfaces son parte de la arquitectura porque están ocultas en el interior de las máquinas y no son visibles desde el exterior. Incluso, tampoco es necesario que las interfaces de todas las máquinas en una red sean las mismas, siempre y cuando cada máquina pueda utilizar correctamente todos los protocolos. La lista de protocolos utilizados por un sistema, un protocolo por capa, se conoce como **pila de protocolos.**

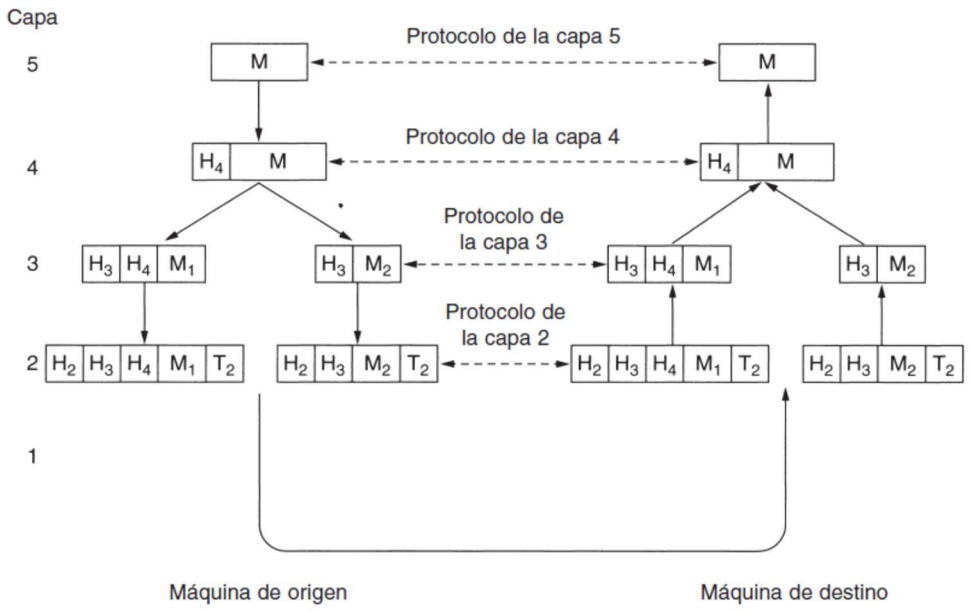


Figura 8.13 - Ejemplo de flujo de información que soporta una comunicación virtual en la capa 5.

Veamos un ejemplo: cómo proporcionar comunicación a la capa superior de la red de cinco capas de la figura 8.13. Un proceso de aplicación que se ejecuta en la capa 5 produce un mensaje, *M,* y lo pasa a la capa 4 para su transmisión.

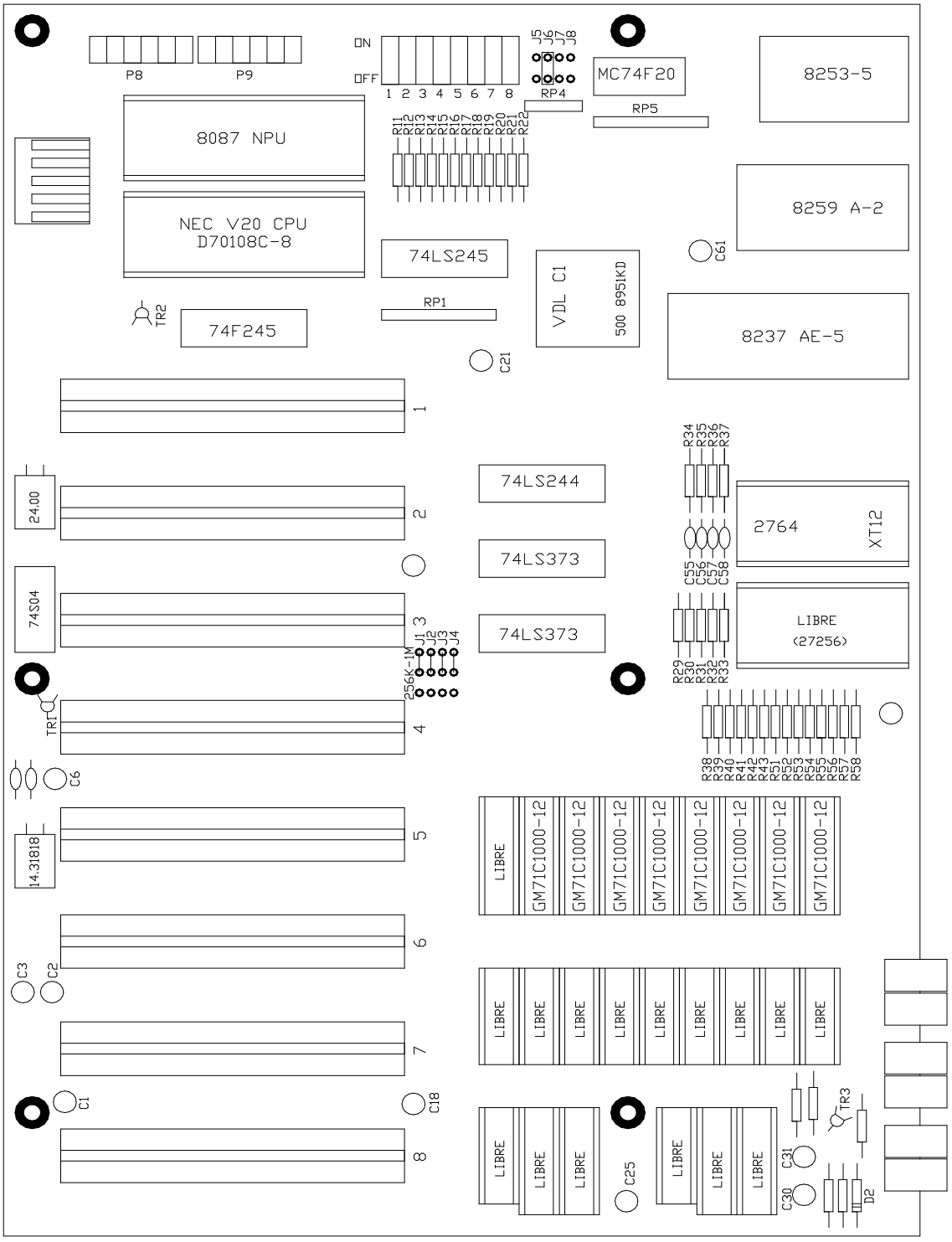
La capa 4 pone un **encabezado** al frente del mensaje para identificarlo y pasa el resultado a la capa 3. El encabezado incluye información de control, como números de secuencia, para que la capa 4 de la máquina de destino entregue los mensajes en el orden correcto si las capas inferiores no mantienen la secuencia. En algunas capas los encabezados también pueden contener tamaños, medidas y otros campos de control.

Bibliografía: Redes de Computadoras – 5ª edición (2013) - Andrew S. Tanenbaum – PEARSON EDUCACION

APÉNDICES

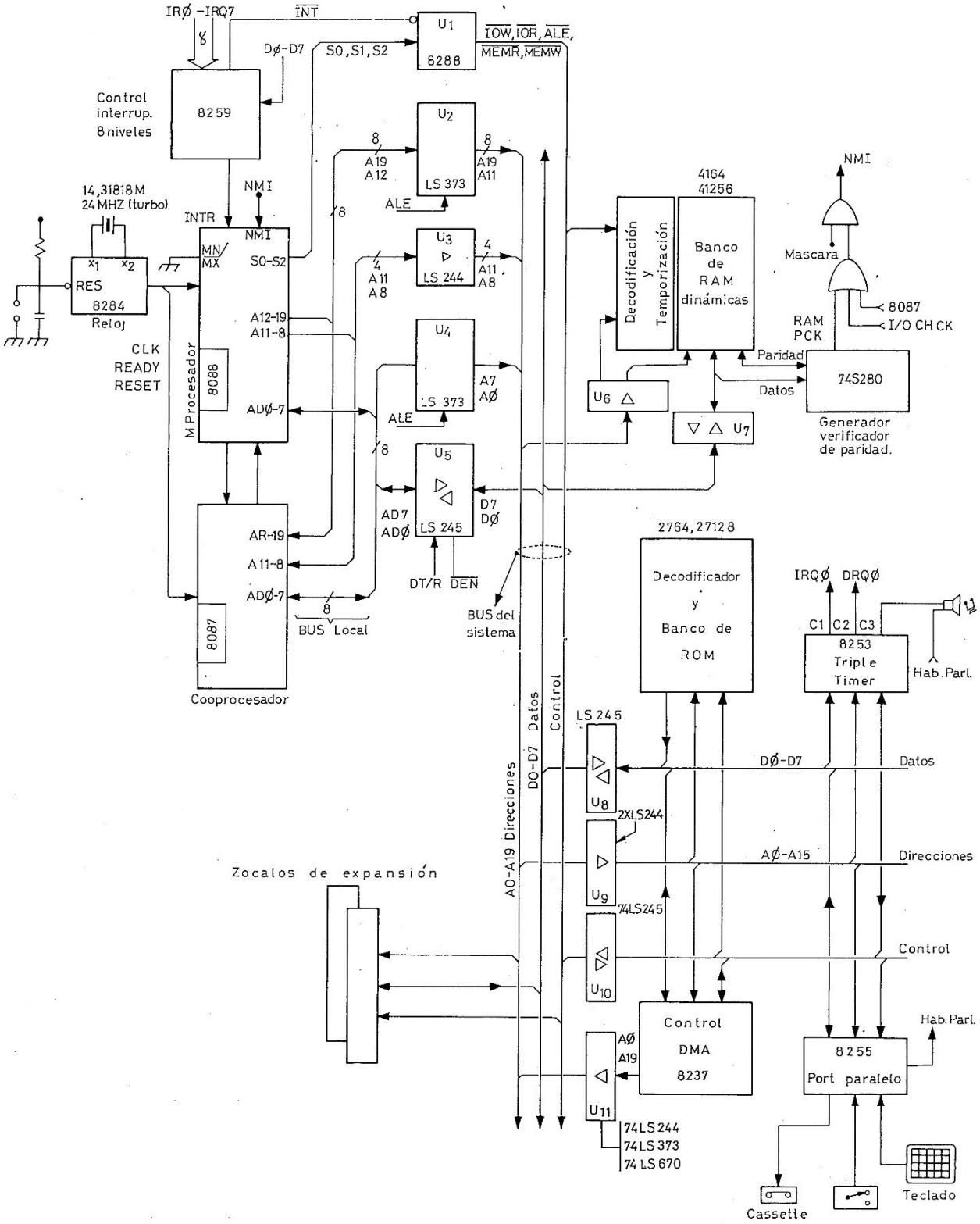
APÉNDICE A - MOTHERBOARDS DE PC

A.1 MOTHER BOARD DE PC-XT – Socket DIL 64



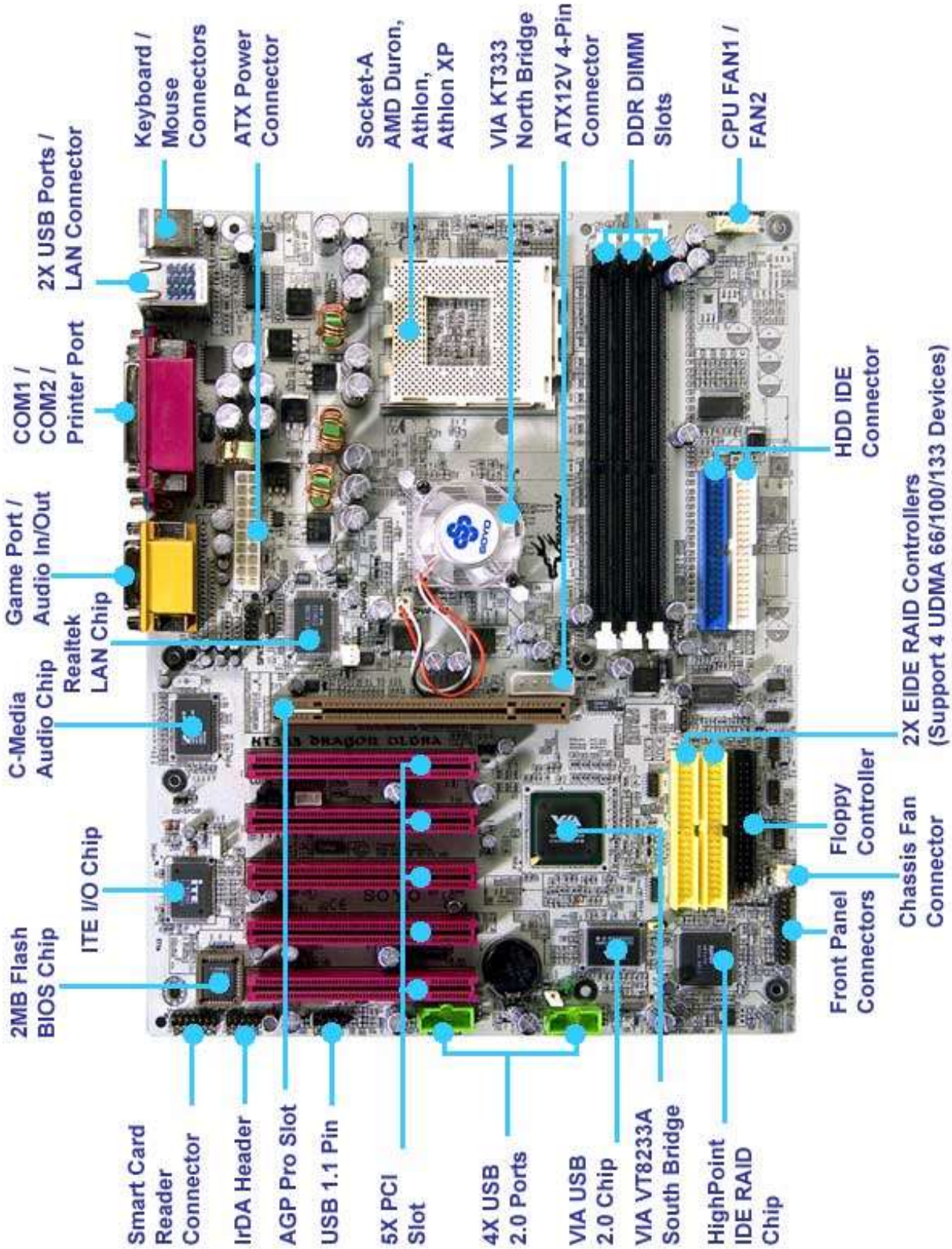
© Ing. Felipe F. Poblete – Dibujo con Autocad ® de mi primera PC, donada a UTN - FRSN

A.2 ARQUITECTURA DEL SISTEMA - PC-XT

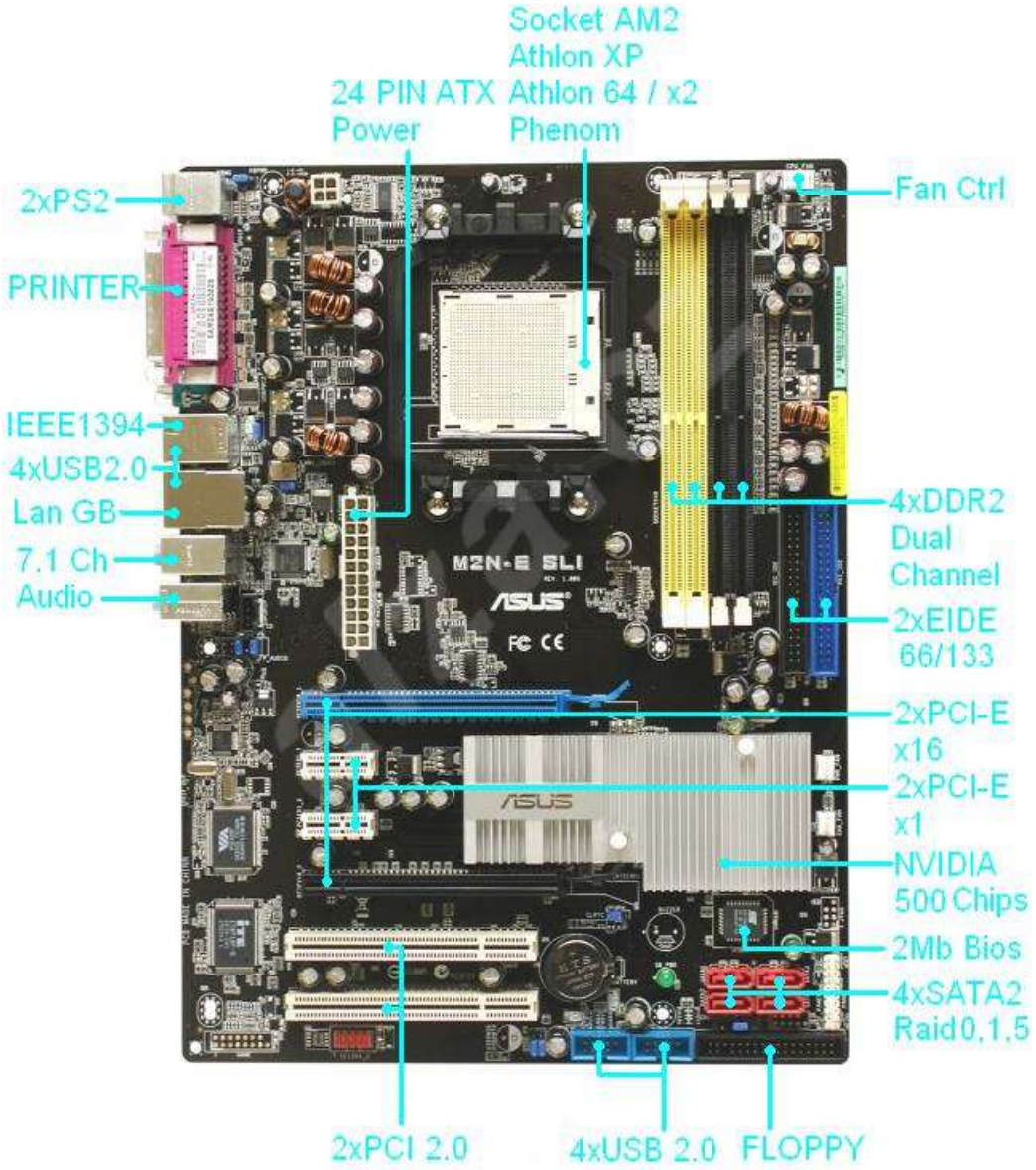


© Revista Telegráfica Electrónica – Estructura interna PC-XT – Autor: Ing. Pascual Carravetta – UTN FRBA - 1987

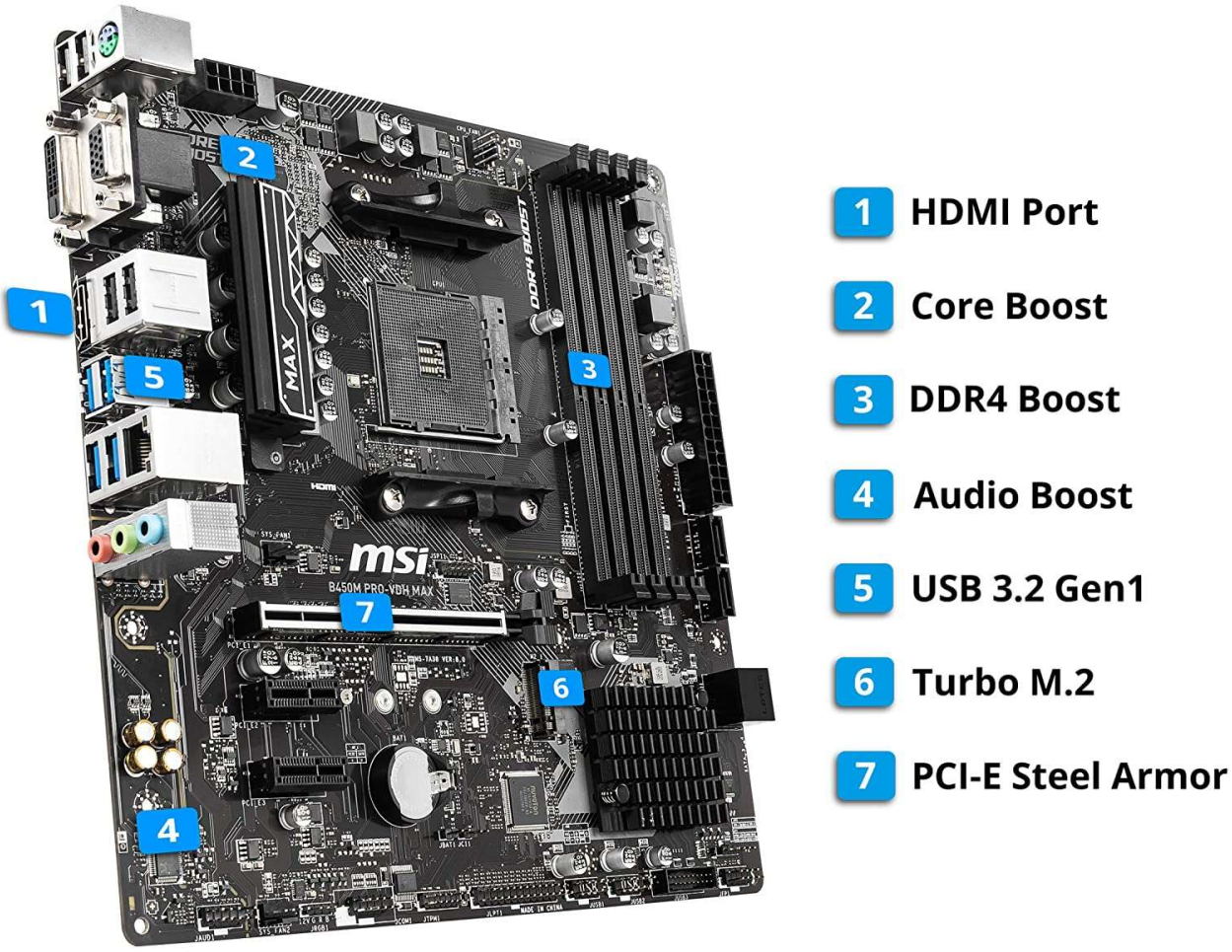
A.3 MOTERBOARD DE PC ATX – Socket A (AMD)

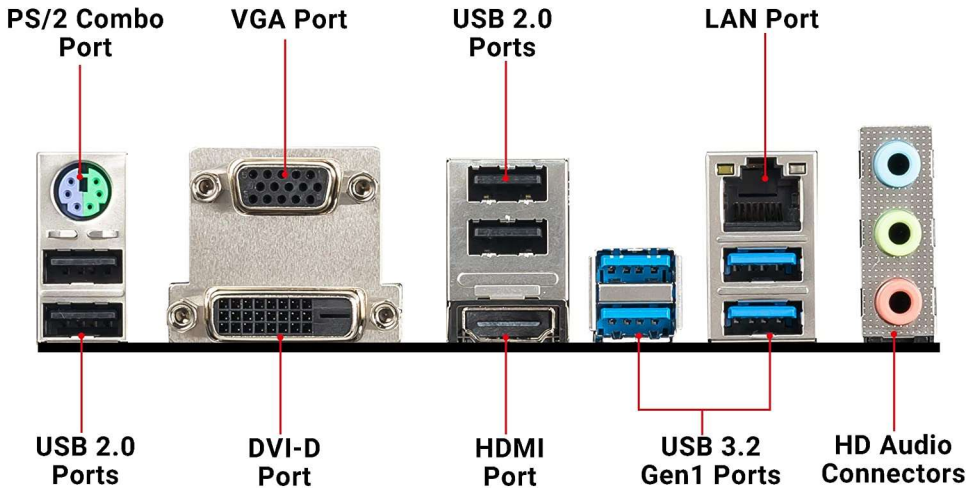


A.4 MOTERBOARD DE PC ATX – Socket AM2 (AMD)

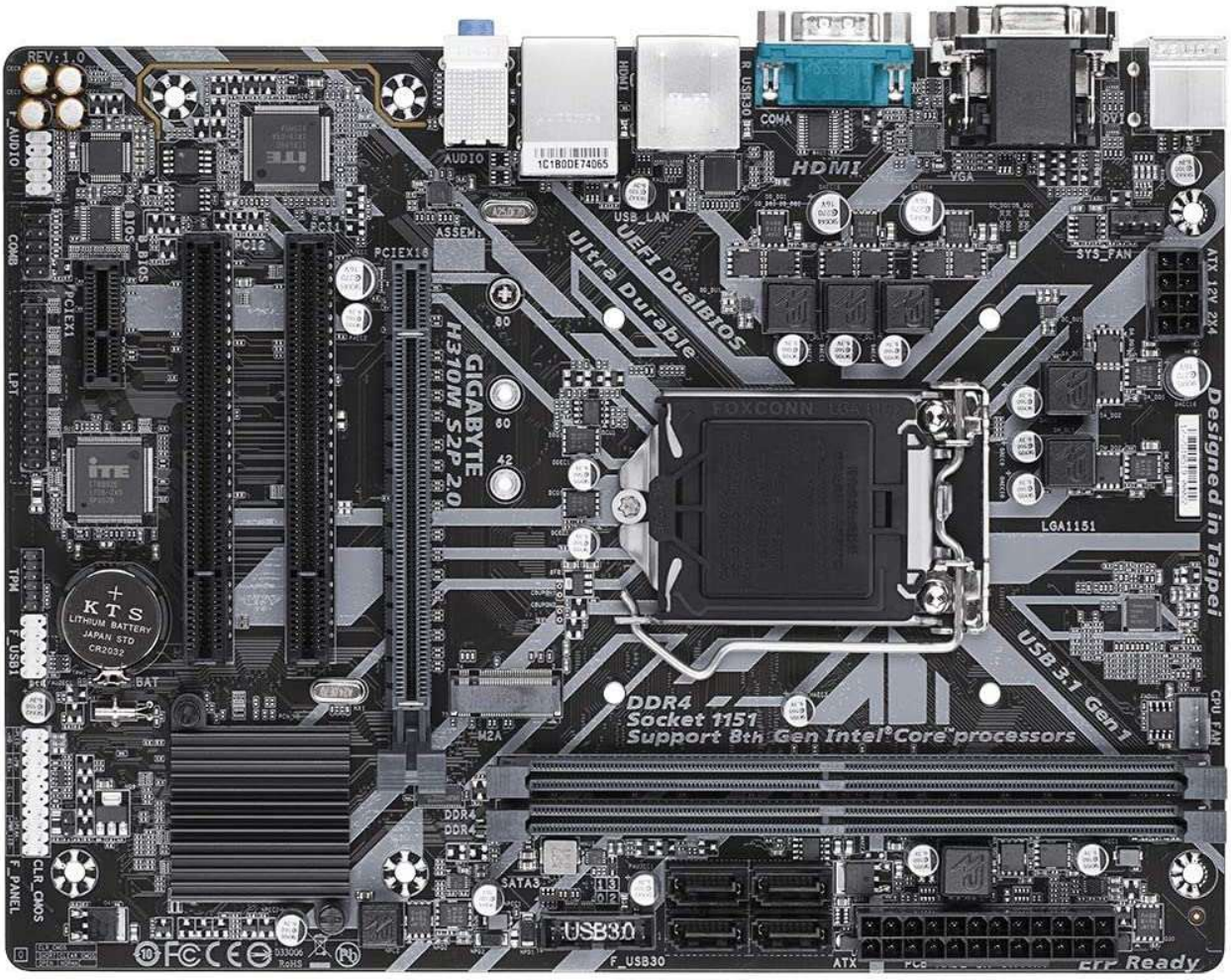


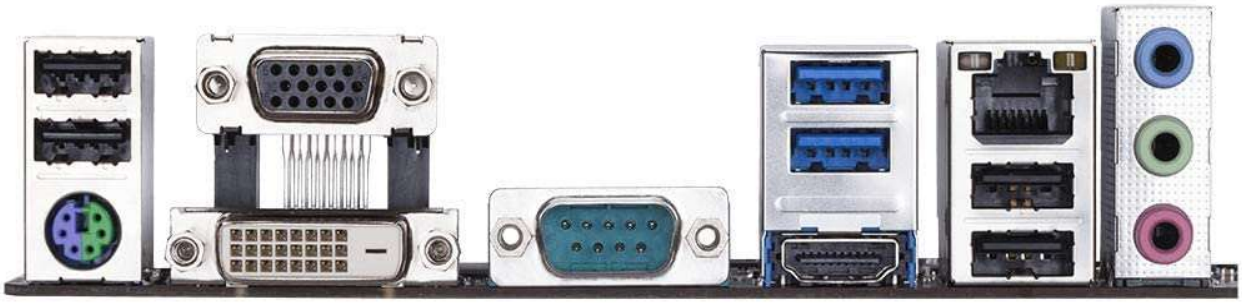
A.5 MOTERBOARD DE PC ATX – Socket AM4 (AMD)





A.6 MOTERBOARD DE PC ATX – Socket LGA1151 (Intel)





APÉNDICE B - DIRECTIVAS DEL ASM86

Directiva SEGMENT / ENDS

Sintaxis:

```
nombre SEGMENT [alineación] [combinación] [nombre_clase]
            PARA      PUBLIC
            BYTE      COMMON
            WORD      STACK
            PAGE      MEMORY
            INPAGE    AT ...
      ...
nombre ENDS
```

Se usa para definir un segmento lógico. Este segmento puede combinarse con otros en el mismo módulo y/o con segmentos definidos en otros módulos. Estos formarán luego los segmentos físicos en memoria, direccionados por los registros de segmento. El programador pondrá entre las directivas SEGMENT y ENDS el código, los datos o la pila.

Directiva ASSUME

Sintaxis:

```
ASSUME reg.segm : núm.parágr [,...:...]
      CS      Nombre de Segmento Ej. CODIGO
      DS      Nombre de grupo Ej. CODGRUPO
      SS      Expresión SEG_nombre
      ES      NOTHING
```

Informa al ensamblador de cuál será el contenido de los registros de segmento en el momento de la ejecución. No inicializa los registros de segmento. Ayuda al ensamblador a asegurarse que son direccionables el código o los datos. Ayuda al ensamblador para la generación automática del prefijo SEGMENT OVERRIDE.

Ejemplo:

```
DATOS SEGMENT                ; define un segmento para datos
  ...
  ABYTE      DB    0         ; algún dato
  ...
DATOS ENDS                   ; fin de la definición del segmento
```

```
PROG SEGMENT                    ; define un segmento para código
  ...
  MOV   AX,DATOS                ; AX = dirección base del segmento
  MOV   DS,AX                   ; inicializa DS (segmento de datos)
  ...
PROG ENDS                       ; fin de la definición del segmento

ASSUME CS:PROG,DS:DATOS         ; asocia PROG al segmento de código
                                ; y DATOS al segmento de datos
```

Directiva GROUP

Sintaxis:

nombre GROUP núm.parágr [,...]

Nombre de Segmento Ej. CODIGO

Expresión SEG_nombre

Se usa para combinar varios segmentos lógicos en un segmento físico. La suma de las extensiones no debe superar 64 kB.

Ejemplo: CODGRUPO GROUP CODIGO1, CODIGO2

Variables, rótulos y constantes

Variables

Los dos objetos mas referidos (a excepción de los registros) son las variables y los rótulos. Las variables son áreas de con nombre donde se almacenan valores. Los rótulos se refieren a secciones de código a donde se puede saltar o llamar. Una variable es definida por una sentencia de definición de datos o una directiva *LABEL*. Cada variable tiene tres atributos:

1. Segmento - El segmento en el cual fue definida. Es un valor que representa el número de parágrafo del segmento.
2. Desplazamiento - Es un valor de 16 bits que indica la distancia en bytes desde el comienzo del segmento hasta la posición de la variable en memoria.
3. Tipo - El tamaño en bytes, usualmente expresado por una palabra clave en la definición. Pueden ser: *BYTE, WORD, DWORD, QWORD, TBYTE*, estructura y registro.

Rótulos

Los rótulos definen direcciones para las instrucciones ejecutables, representando un "nombre" para una locación en el código. Un rótulo puede ser definido de tres formas: 1) un

nombre seguido de ":" asociado con una sentencia de instrucción, 2) una directiva *PROC*, o 3) una directiva *LABEL*. Como las variables, los rótulos tienen tres atributos:

1. Segmento - Ídem que las variables.
2. Desplazamiento - Ídem que las variables.
3. Tipo - Especifica el tipo de salto o llamado a subrutina que debe ser hecho a esa locación. Puede ser *NEAR* (dentro del mismo segmento) o *FAR* (desde otro segmento).

Ej.: LAZO: JMP COMIENZO ; este rótulo será NEAR

Constantes

Son números puros, sin ningún atributo. Una constante puede ser un número binario, octal, decimal, hexadecimal, ASCII, real decimal o real hexadecimal. Se pueden evaluar en 8 bits, 16 bits o real. Ejemplos: 11011B, 174Q, 983[D], F7DH, 'texto', 3.1416 o .002E7 y 40490FDBR.

Definición e inicialización de variables

Directiva DB

Sintaxis:

[nombre] DB valor [,...]

Constante decimal o entera (<255)

?

Cadena de caracteres

repeticiones DUP valor

Inicialización de un byte.

Directiva DW

Sintaxis:

[nombre] DW valor [,...]

Constante decimal o entera (<65535)

?

Variable o expresión

repeticiones DUP valor

Inicialización de una palabra (2 bytes).

Directiva DD

Sintaxis:

[nombre] DD valor [,...]

Constante entera ($<2^{32}-1$) o real (<E±38)

?

Variable o expresión

repeticiones DUP valor

Inicialización de una palabra doble (4 bytes).

Directiva DQ

Sintaxis:

[nombre] DQ valor [,...]

Constante entera (<2^{64}-1) o real (<E±308)

?

Variable o expresión

repeticiones DUP valor

Inicialización de una palabra cuádruple (8 bytes).

Directiva DT

Sintaxis:

[nombre] DT valor [,...]

Constante entera BCD (<10^{18}-1), real (<E±4932)

?

Variable o expresión

repeticiones DUP valor

Inicialización de 10 bytes.

Ejemplos:

1 - El operador DUP se puede usar para reservar espacio de memoria sin inicializar.

```
PILA SEGMENT STACK          ; define un segmento para la pila
  ...
        DW 128 DUP (?)      ; reserva 128 palabras de 16 bits
  ...
PILA ENDS                   ; fin de la definición del segmento
```

2 - Los textos se pueden ingresar directamente en ASCII

```
  CARTEL      DB 'NUMEROS, CONVERSION...$'
```

Directiva RECORD

Sintaxis: (plantilla)

nombre *RECORD* nombre_campo : expresión [=val_inicial] [,...]

Sintaxis: (inicialización)

nombre nombre_record <[expresión] [,...]>

nombre nombre_record repeticiones DUP (<[expresión] [,...]>)

El ASM86 tiene una sentencia de inicialización de datos especial que permite construir estructuras de datos orientadas a bits, llamadas *records*, que pueden tener 8 o 16 bits de tamaño. Cada record se define con un número de campos conteniendo un cierto número de bits por campo. Se puede almacenar información en esos campos y también acceder a esa información. Los records son útiles cuando se desea acceder a bits específicos en una estructura de datos. Estos pueden ser indicadores (flags) o los campos usados para almacenar un número real.

Para utilizar un record, primero hay que definir la plantilla (template) que especifica el tamaño del record y sus campos. Luego se usa el nombre del record en una sentencia de inicialización de datos para asignar el almacenamiento.

Ejemplo de plantilla:

ERRORFLAGS *RECORD* IOERR:3=0, SYSERR:3=0, MEMERR:3

Ejemplo de inicialización:

FLAGS ERRORFLAGS <0,3,0> ; sobrescribe las asignaciones de la plantilla

Directiva PROC

Sintaxis:

nombre PROC [tipo]

NEAR (valor por defecto)

FAR

nombre ENDP

Se usa para definir un rótulo, que normalmente comprende una serie de instrucciones y es referido como una subrutina. las instrucciones que se encuentren fuera del par PROC / ENDP no serán ejecutadas en el llamado a "nombre".

Directiva LABEL

Sintaxis:

nombre LABEL [tipo]

BYTE

WORD

DWORD

TBYTE

QWORD

nombre de estructura

nombre de record

NEAR (valor por defecto)

FAR

La directiva LABEL crea un nombre para la locación actual del ensamblado, ya sean datos o código. La directiva LABEL define un rótulo o variable que tendrá los atributos que fueran tratados en definición de rótulos.

Puede ser usada para nombrar la misma posición con dos nombres diferentes.

Ejemplo:

```
AWORD     LABEL        WORD
LOWBYTE   DB      0
HIGHBYTE  DB      0
```

Se puede acceder a la palabra completa con AWORD o a cada uno de los bytes con LOWBYTE o HIGHBYTE.

Contador de locaciones

El contador de locaciones sigue el desplazamiento actual dentro del segmento actual que está siendo ensamblado. este valor, simbolizado por el carácter $ puede ser usado en ciertos contextos (por ejemplo: expresiones o instrucciones). Este símbolo representa un rótulo NEAR.

Directiva ORG

Sintaxis:

ORG expresión

La directiva ORG permite controlar el contador de locaciones dentro del segmento actual. Se usa para fijar el contador de locaciones al valor deseado. Usada con un segmento absoluto, se puede especificar la locación de memoria en la cual el código o los datos serán localizados.

Ejemplo:

```
ORG OFFSET ( $ + 1000 )
```

Directiva EVEN

Sintaxis:

EVEN

El código o los datos que siguen a la directiva serán alineados en límite de palabras, logrando mayor velocidad en el 8086. El ensamblador colocará una instrucción de no operación NOP (90H) si es necesario.

Directiva PURGE

Sintaxis:

PURGE nombre

Borra la definición de un símbolo, permitiendo su re definición.

Directivas de vinculación de programas.

El ASM86 suministra las directivas necesarias para soportar programas multi-modulares. Un programa puede estar compuesto por varios módulos individuales (ASM(&, PL/M86, PASCAL86, FORTRAN86) los cuales son ensamblados o compilados individualmente. Cada módulo puede definir variables que los otros pueden usar. El mecanismo para comunicar información de símbolos entre un módulo y otro son las directivas *EXTRN* y *PUBLIC*.

Directiva PUBLIC

Sintaxis:

PUBLIC nombre [,...]

La directiva PUBLIC especifica que símbolos en el módulo son accesibles a otros módulos en el momento de la vinculación. Estos símbolos pueden ser variables, rótulos o constantes.

Directiva EXTRN

Sintaxis:

EXTRN nombre : tipo [,...]

La directiva EXTRN especifica aquellos símbolos que serán referidos en el módulo y que han sido declarados PUBLIC otro módulo. La directiva EXTRN indicará el nombre del símbolo y su tipo (véase **tipo** en la explicación correspondiente a la directiva LABEL).

La directiva EXTRN se debe colocar en el mismo segmento en el que se definió como PUBLIC en el otro módulo. Si no se conoce en que segmento fue definido PUBLIC, se debe poner EXTRN fuera de todos los pares SEGMENT / ENDS del programa.

Directiva END

Sintaxis:

END [rótulo]

La directiva END es necesaria en todos los módulos de programa ASM86 y es la última sentencia del módulo. Ella determina la finalización del proceso de ensamblado; cualquier texto que se encuentre después no será tenido en cuenta.

Otro propósito de la directiva END es definir un módulo como principal (MAIN-MODULE). Esto quiere decir que contiene el código que será ejecutado primero, cuando se cargue el programa en memoria. La ejecución comenzará a partir del rótulo indicado como dirección de arranque en la directiva END. Como opcional, puede usarse para definir el contenido inicial de los registros de segmento DS y SS; en este caso debe especificarse también el par CS:IP.

Directiva NAME

Sintaxis:

NAME nombre de módulo

La directiva NAME se utiliza cuando se vincularán a la vez distintos módulos.

Operadores y expresiones

Expresiones de direccionamiento

```
ADD   DX,COUNT          ; DX se suma al valor contenido en COUNT
ADD   DX,COUNT + 2      ; DX se suma al valor contenido en COUNT + 2
ADD   DX,COUNT [2]      ; ídem anterior (otra sintaxis)
```

Operadores aritméticos

HIGH, LOW (byte alto, byte bajo)

```
Ej.:    MOV   AH,HIGH(1234H)          ; AH=12H
Ej.:    MOV   TENHEX EQU  (0FF10H)    ; TENHEX=10H
```

**, /, +,* - (multiplicación, división, suma, resta)

```
Ej.:    CMP   AL,2*4          ; compara AL con 8
Ej.:    MOV   CX,123H/16      ; CX=12
```

SHR, SHL (desplazamiento a la derecha, desplazamiento a la izquierda)

```
Ej.:    MOV   BX,0FACBH  SHR 4          ; BX = FACH
```

Operadores relacionales

EQ igual

NE no es igual

LT menor

LE menor o igual

GT mayor

GE mayor o igual

Ej.: MOV AL,3 EQ 0 ; AL = 0 (falso)

Ej.: MOV BX,2 LE 15 ; BX = 0FFFFH (verdadero)

Operadores Lógicos

OR, XOR, AND, NOT

Operadores para forzar atributos

Ignorar segmento por defecto (segment override)

Sintaxis:

CS: variable

DS: variable

ES: variable

SS: variable

nombre seg.: variable

nombre grupo: variable

1 - Tiene precedencia sobre la sentencia ASSUME pero sólo para esa línea de programa.

2 - Tiene precedencia sobre la asignación de segmentos por defecto para una instrucción.

Ejemplos:

1 -

```
ASSUME DS:DATA, CS:CODE
DATA   SEGMENT
ABYTE DB      0
DATA ENDS
CODE   SEGMENT
  MOV  BL,ABYTE          ; la referencia es cubierta por ASSUME
  MOV  BL,ES:ABYTE       ; no se requiere ASSUME para esta referencia
CODE ENDS
```

2 -

```
  MOV  BL,[BX]           ; en este caso se usará el segmento de datos (DS)
  MOV  BL,ES:[BX]        ; en este caso se usará el segmento extra (ES)
```

Operador PTR

Sintaxis:

tipo PTR nombre

BYTE

WORD

DWORD

QWORD

TBYTE

NEAR

FAR

Define una referencia a memoria de un tipo en particular, para que el programa ensamblador seleccione la instrucción correcta.

Ejemplos:

```
MOV   WORD PTR   [BX], 5          ; en la palabra apuntada por BX pone 5
MOV   CL, BYTE  PTR AWORD    ; lee el primer byte de AWORD
MOV   DL, BYTE  PTR AWORD+1          ; lee el segundo byte de AWORD
```

Operador SHORT

```
Ej.:      JMP    FWDLAB                    ; instrucción de 3 bytes
Ej.:      JMP    SHORT  FWDLAB             ; instrucción de 2 bytes
```

Operador THIS

Define la posición de memoria actual, del ***tipo*** indicado.

Ej.: AWORD EQU *THIS WORD*

Nota: los siguientes operadores tienen la sintaxis: ***OPERADOR* variable**

Operador SEG

Toma el valor del segmento en el que fue definida la variable.

Ej.: ASSUME CS: *SEG* START

Operador OFFSET

Toma el valor del desplazamiento (desde el comienzo del segmento) que tiene la variable. En definitiva: es la dirección relativa.

```
Ej.:      MOV   BX, OFFSET ASCIITABLE          ; es equivalente a
Ej.:      LEA    BX, ASCIITABLE
```

Operador TYPE

Devuelve un valor acorde a la cantidad de bytes de ese tipo de variable.

Ej.: byte = 1 ; dword = 4

Operador LENGTH

Devuelve un valor acorde a la cantidad de unidades de datos (bytes, words, etc.) asignados a un nombre de variable.

Ej.: AWORDARRAY DW 150 DUP (0) ; *LENGTH* = 150

Operador SIZE

Devuelve un valor acorde a la cantidad de bytes asignados a esa variable.

Ej.: AWORDARRAY DW 150 DUP (0) ; *SIZE* = *LENGTH* * *TYPE* = 150 * 2 = 300

Directiva EQU

Sintaxis:

```
nombre      EQU   valor
```

Asigna un valor/expresión numérico o un registro a un nombre que resulte mas mnemotécnico, sin asignar posiciones de memoria.

```
Ej.:    COUNT       EQU   CX
                ...
                MOV   COUNT, 10          ; CX = 10

Ej.:    MOVER       EQU   MOV
                ...
                MOVER  AX, BX
```

APENDICE C - TURBO DEBUG

La siguiente pantalla muestra la ejecución en TD.EXE del programa explicado en el capítulo 3 (ensayo.exe).

```
Símbolo del sistema - td ensayo
File   View   Run   Breakpoints   Data   Window   Options                READY
CPU 80386
cs:0000►B88C44        mov    ax,448C        ax 0000    c=0
cs:0003 8ED8          mov    ds,ax          bx 0000    z=0
cs:0005 B89144        mov    ax,4491        cx 0000    s=0
cs:0008 8ED0          mov    ss,ax          dx 0000    o=0
cs:000A 8D260001      lea    sp,[0100]      si 0000    p=0
cs:000E 8D160C00      lea    dx,[000C]      di 0000    a=0
cs:0012 B409          mov    ah,09          bp 0000    i=1
cs:0014 CD21          int    21             sp 00FE    d=0
cs:0016 8D060200      lea    ax,[0002]      ds 447C
cs:001A 50            push   ax             es 447C
cs:001B A10000        mov    ax,[0000]      ss 4491
cs:001E 50            push   ax             cs 44A1
cs:001F E80E00        call   0030           ip 0000

ds:0100 05 00 01 00 02 00 03 00 ♣ ♫ ☻ ♥
ds:0108 04 00 05 00 54 45 43 4E ♦ ♣ TECN
ds:0110 49 43 41 53 20 44 49 47 ICAS DIG
ds:0118 49 54 41 4C 45 53 20 49 ITALES I

ss:0100 8CB8
ss:00FE►0000

F2-Bkpt F3-Close F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Luego de la ejecución, se puede ver la tabla de datos ordenada.

```
Símbolo del sistema - td ensayo
File   View   Run   Breakpoints   Data   Window   Options                READY
CPU 80386
44A1:0000 B88C44      mov    ax,448C        ax 06F7    c=1
44A1:0003 8ED8        mov    ds,ax          bx B25E    z=0
44A1:0005 B89144      mov    ax,4491        cx 31DF    s=1
44A1:0008 8ED0        mov    ss,ax          dx 62C5    o=0
44A1:000A 8D260001    lea    sp,[0100]      si B242    p=0
44A1:000E 8D160C00    lea    dx,[000C]      di 0200    a=0
44A1:0012 B409        mov    ah,09          bp 0100    i=1
44A1:0014 CD21        int    21             sp 0106    d=1
44A1:0016 8D060200    lea    ax,[0002]      ds 31DF
44A1:001A 50          push   ax             es B27E
44A1:001B A10000      mov    ax,[0000]      ss 06F7
44A1:001E 50          push   ax             cs 0000
44A1:001F E80E00      call   0030           ip 0000

447C:0100 05 00 05 00 04 00 03 00 ♣ ♣ ♦ ♥
447C:0108 02 00 01 00 54 45 43 4E ☻ ♫ TECN
447C:0110 49 43 41 53 20 44 49 47 ICAS DIG
447C:0118 49 54 41 4C 45 53 20 49 ITALES I

4491:0100 8CB8
4491:00FE 3202

F2-Bkpt F3-Close F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

APÉNDICE D – LEY DE MOORE

En 1965, cuando en un CI se podían integrar 30 transistores, Gordon Moore dijo que "la cantidad de transistores se duplicaría cada 18 meses". Luego, en 1975 lo modificó a 24 meses".

Hoy, cinco décadas después, la aseveración se cumple con bastante aproximación, incluso para otros dispositivos, como los discos rígidos.

Fig 1 - Gordon Moore – vicepresidente y co-fundador de Intel Corp.

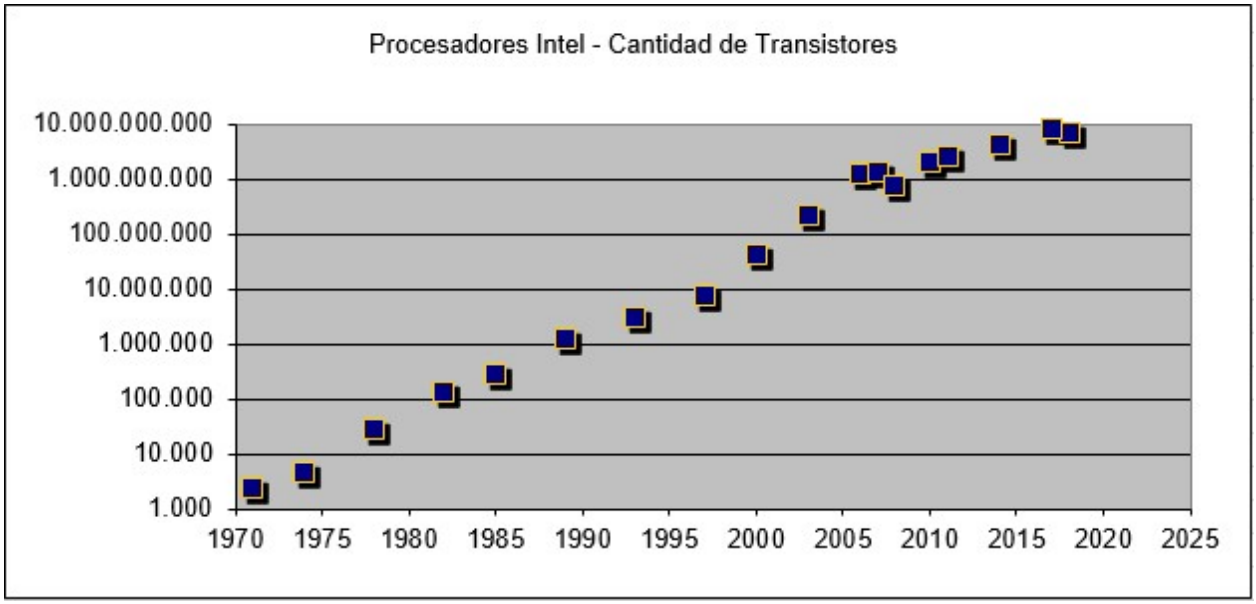


Fig 2 – Microprocesadores

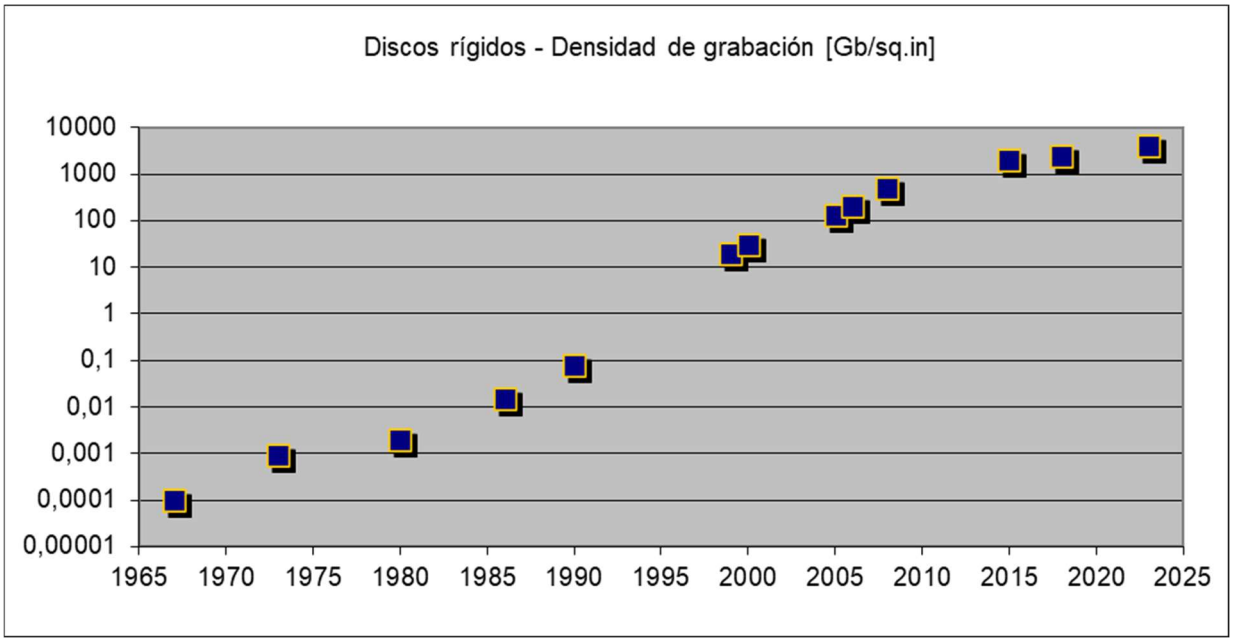


Fig 3 – Discos rígidos

APÉNDICE E – CARACTERÍSTICAS DE LOS MICROPROCESADORES Y DISCOS RÍGIDOS

Tabla 1 – Microprocesadores Intel

Año	Modelo	Bus Datos	Bus Direcc	Transistores	Tecnología [nm]	Frecuencia [MHz]	Modo prot	Mem virtual	Core	Cache L1 [KB]	Cache L2 [KB]	Cache L3 [KB]	Etapas pipeline	Unid. ejec.	V core	Otros
1971	4004	4	12	2.300	10000	0,108	no	no	1				1			
1974	8080	8	16	4.500	3000	1	no	no	1				1		5	
1978	8086	16	20	29.000	3000	4,7	no	no	1				2		5	
1982	80286	16	24	134.000	1500	10	si	no	1							
1985	80386	32	32	275.000	1000	33	si	si	1				4			
1989	80486DX	32	32	1.200.000	1000	50	si	si	1	8						NPX
1993	Pentium	32 (64)	32	3.100.000	800	200	si	si	1	2x8			5		3,3	
1997	PentiumII	32 (64)	32	7.500.000	350	450	si	si	1	2x16	512					MMX
2000	Pentium4	64	36	42.000.000	180	1500	si	si	1	2x16	1000					HT
2003	Itanium 2	64	36	220.000.000	90	1600	si	si	2	4x16	2x256	6000				HT
2006	Core2Quad	64	36	1.200.000.000	65	2600	si	si	4	4x32	2x4000					VT
2007	Quad Core Xeon	64	36	1.330.000.000	45	3200	si	si	4	8x32	2x6000					VT
2008	Core I7-960	64	36	731.000.000	45	3200	si	si	4	8x32	4x256	8000	14			HT(2),QPI
2010	Itanium 9350	64	50	2.000.000.000	45	1866	si	si	4	8x16	4x768	4x6000*	8 (EPIC)			HT(2),VT-i2
2011	Xeon Westmere	64	50	2.600.000.000	32		si	si	10							
2014	Xeon E7 V2	64		4.310.000.000	32	3400	si	si	15		15x1000	37500		30		
2018	CoreI9-7980XE	64	46	7.000.000.000	14	4300	si	si	18	18x64	18x1000	24750				HT
2017	Xeon 8180M	64		8.000.000.000	14	3800	si	si	28	28x64	28x1000	38500				HT

Fuente: intel.com y otros

Tabla 2 – Microprocesadores AMD

Año	Modelo	Bus Datos	Bus Direcc	Transistores [M]	Tecnología [nm]	Frecuencia	Modo prot	Mem virtual	Core	Cache L1 [KB]	Cache L2 [KB]	Cache L3 [KB]	V core	TDP [W]	Otros
1993	AM486-DX	32 (64)	32	1		25	si	si	1	8					
1996	K5-100 PR133	32 (64)	32	4	350	100	si	si	1	16+8			3,52		
1997	K6 300	32 (64)	32	9	250	300	si	si	1	2x32			2,9		MMX
1998	K6-2+ 500	64	36	15	180	500	si	si	1	2x32	128		2		
2000	K7 Athlon	64	40	37	180	1000	si	si	1	2x64	256		1,75		3Dnow
2002	Athlon XP	64	40	54	130	3000	si	si	1	2x64	512		1,65		
2004	Opteron	64	40	70	130	2000	si	si	2	4x32	2x512		1,2		HT (2)
2006	Athon64x2 6000	64	40	200	90	3000	si	si	2	4x32	2x1000		1,35		
2007	Phenom	64	40	450	65	2200	si	si	4	8x64	4x512	2000	1,1		VT
2009	Opteron 2400	64		904	45		si	si							
2011	FX	64		1.200	32		si	si	8						
2013	Opteron 6300	64		2.700	32	2800	si	si	16		16x1000	16000			
2016	Ryzen 7 1800X	64		4.800	14	4000	si	si	8	8x96	8x512	16000			
2017	EPYC 7601	64		19.200	14	2700	si	si	32	32x96	32x512	64000			
2019	Ryzen 9 3900X	64		9.890	12										
2023	Epyc Bergamo	64		82.000	6	3100	si	si	128			256000		360	HT (2)

Fuente: amd.com y otros

Tabla 3 – Discos rígidos

Año	Modelo	Densidad [Gb/sq.in]	Capacidad [MB]	Velocidad [MB/s]	Tecnología de grabación	Platos	Diámetro [inch]	MTBF [Mh]
1967	Hitachi H-8564	0,0001	7,5		horizontal	6	14	
1973	Hitachi H-8589	0,0009	100	0,8	horizontal - MFM	10	14	
1980	Seagate ST 506	0,002	5		horizontal		5,25	
1986	Fujitsu	0,015	26		horizontal			
1990	Hitachi DK 314	0,08	418	2,0	horizontal	7	3,5	
1999	Fujitsu	20	100000		horizontal- GMR			
2000	Seagate Cheetah	30	180000		horizontal			
2005	Toshiba	133			perpendicular			
2006	Seagate Barracuda	200	750000		perpendicular			
2008		500	1000000	70,0	perpendicular			
2015	Western digital	2000	8000000	120,0	perpendicular	4	3,5	
2018	Seagate	2500	14000000	130,0	perpendicular	8	3,5	
2023	Western digital	4000	22000000	291,0	perpendicular	4	3,5	2,5

Fuente: seagate.com, maxtor.com, toshiba.com, hitachi.com y otros

APÉNDICE F – ESTADÍSTICAS

PassMark - CPU Mark

High End CPUs

Updated 13th of February 2024

	CPU Mark	Price (USD)
AMD Ryzen Threadripper PRO 7995WX	153,962	$9,999.00*
AMD Ryzen Threadripper PRO 7985WX	136,214	NA
AMD Ryzen Threadripper 7980X	133,905	NA
AMD EPYC 9654	126,045	$6,690.00
AMD EPYC 9654P	113,949	$10,625.00*
AMD EPYC 9554P	109,858	$7,104.00*
AMD EPYC 9474F	104,894	NA
AMD Ryzen Threadripper 7970X	99,129	NA
AMD Ryzen Threadripper PRO 7975WX	97,474	NA
AMD EPYC 9454P	96,088	$4,598.00*

Fig. 1 – Top ten de procesadores

Fuente: https://www.cpubenchmark.net/high_end_cpus.html

PassMark - Price Performance (CPU Mark / $Price)

Top 150 Available Price Performance CPUs

Updated 13th of February 2024

	CPU Value (Mark/Price)	CPU Mark	Price (USD)
AMD Ryzen 5 5500	229.3	19,489	$84.99
AMD Ryzen 5 4500	215.2	16,158	$75.09
AMD Ryzen 5 3600	212.9	17,775	$83.49
AMD Ryzen 5 5600	179.9	21,586	$119.99
AMD Ryzen 3 4100	170.5	11,081	$64.99
AMD Ryzen 5 5600G	163.4	19,900	$121.80
AMD Ryzen 3 3100	162.2	11,622	$71.65
AMD Ryzen 5 4600G	157.0	16,015	$101.99
AMD Ryzen 7 3700X	153.7	22,591	$146.97
Intel Core i5-13400F	153.3	25,293	$164.99

Fig. 2 – CPU - Mejor relación Performance vs Precio

Fuente: https://www.cpubenchmark.net/cpu_value_available.html

PassMark - G3D Mark

High End Videocards

	Average G3D Mark	Price (USD)
GeForce RTX 4090	38,813	1,799.99
GeForce RTX 4080 SUPER	34,676	999.99
GeForce RTX 4080	34,634	1,111.68
GeForce RTX 4070 Ti SUPER	31,984	799.99
GeForce RTX 4070 Ti	31,743	739.99
Radeon 610M Ryzen 9 7945HX3D	31,600	NA
Radeon RX 7900 XTX	30,949	929.99
GeForce RTX 4070 SUPER	30,673	589.99
GeForce RTX 3090 Ti	29,806	1,539.99*
Radeon PRO W7900	29,501	NA

Fig. 3 – Top ten de placas de video

Fuente: https://www.videocardbenchmark.net/high_end_gpus.html

PassMark - Price Performance (G3D Mark / $Price)

Top 50 Available Price Performance Videocards

	G3D Mark / Price	G3D Mark	Price (USD)
Radeon RX 6600M	71.7	13619	189.99
GeForce RTX 4060	66.7	19689	294.99
GeForce RTX 3060 Ti	64.3	20585	319.99
Radeon RX 7600	62.4	16231	259.99
GeForce RTX 2060	60.7	14147	233.13
GeForce RTX 4060 Ti	60.4	22667	374.99
GeForce RTX 3060 12GB	59.1	17105	289.39
GeForce RTX 3050	57.5	12933	224.99
Radeon RX 6500 XT	56.3	9516	169.00
GeForce GTX 1660 SUPER	56.1	12780	227.99

Fig. 4 – GPU - Mejor relación Performance vs Precio

Fuente: https://www.videocardbenchmark.net/gpu_value.html

PassMark - Disk Rating

High End Drives

	Disk Rating	Price (USD)
NVMe CT2000T700SSD5	82,532	NA
T-FORCE TM8FF1002T	76,065	NA
Seagate ST8000NM000A	75,781	NA
Corsair MP700 PRO	73,738	NA
Crucial T700 2TB	73,468	299.99
Crucial T700 4TB Gen5	73,215	499.99
GIGABYTE AG512K2TB	71,952	344.24
Crucial T700 2TB Gen5 M.2 with heatsink	70,018	NA
MSI M570 2TB	68,265	349.99
Crucial T700 2TB Gen5	68,112	NA

Fig. 5 – Top ten de discos rígidos (incluye SSD)

Fuente: https://www.harddrivebenchmark.net/high_end_drives.html

PassMark - Price Performance (Disk Rating / $Price)

Top 50 Available Price Performance Drives

	Disk Rating / Price	Disk Rating	Price (USD)
BC711 NVMe SK hynix 256GB	984.7	16,160	16.41
NVMe 256GB	961.2	12,486	12.99
Micron 2300 MTFDHBA512TDV	807.0	21,346	26.45
HFM256GD3JX013N	774.1	14,624	18.89
Crucial P3 Plus 500GB	767.5	20,715	26.99
Micron 2200S NVMe 256GB	766.5	13,752	17.94
NVMe HFM512GD3JX013N	759.6	21,528	28.34
WD Blue SN580 500GB	749.4	29,967	39.99
WDC PC SN810 SDCPNRY 1TB	747.6	41,115	55.00
Patriot M.2 P400 512GB	739.3	28,085	37.99

Fig. 6 – Drives - Mejor relación Performance vs Precio

Fuente: https://www.harddrivebenchmark.net/hdd_value.html

PassMark - Memory Read Transfer Rate

Top DDR5 Memory Modules

Updated 13th of February 2024

	Transfer Rate	Price (USD)
Essencore Limited KD5AGU880-60A300G 16GB	41,269 MB/s	NA
Kingston 9905782-029.A00G 32GB	41,243 MB/s	NA
Wilk Elektronik S.A. IR-6400D564L32/32G 32GB	41,238 MB/s	NA
Corsair CMK96GX5M4B5600C40 24GB	41,234 MB/s	NA
Kingston 9905782-105.A00G 32GB	41,214 MB/s	NA
Corsair CMP64GX5M2X6000Z30 32GB	40,465 MB/s	NA
A-DATA Technology AX5U5200C3816G-CLARWH 16GB	40,047 MB/s	NA
AMD R5S532G5600U2S 32GB	40,002 MB/s	NA
G Skill Intl F5-6000J3036G32G 32GB	39,683 MB/s	NA
Kingston KF560C32-48 48GB	39,668 MB/s	NA

Fig. 7 – Top ten de memoria RAM

Fuente: https://www.memorybenchmark.net/read_uncached_ddr5.html

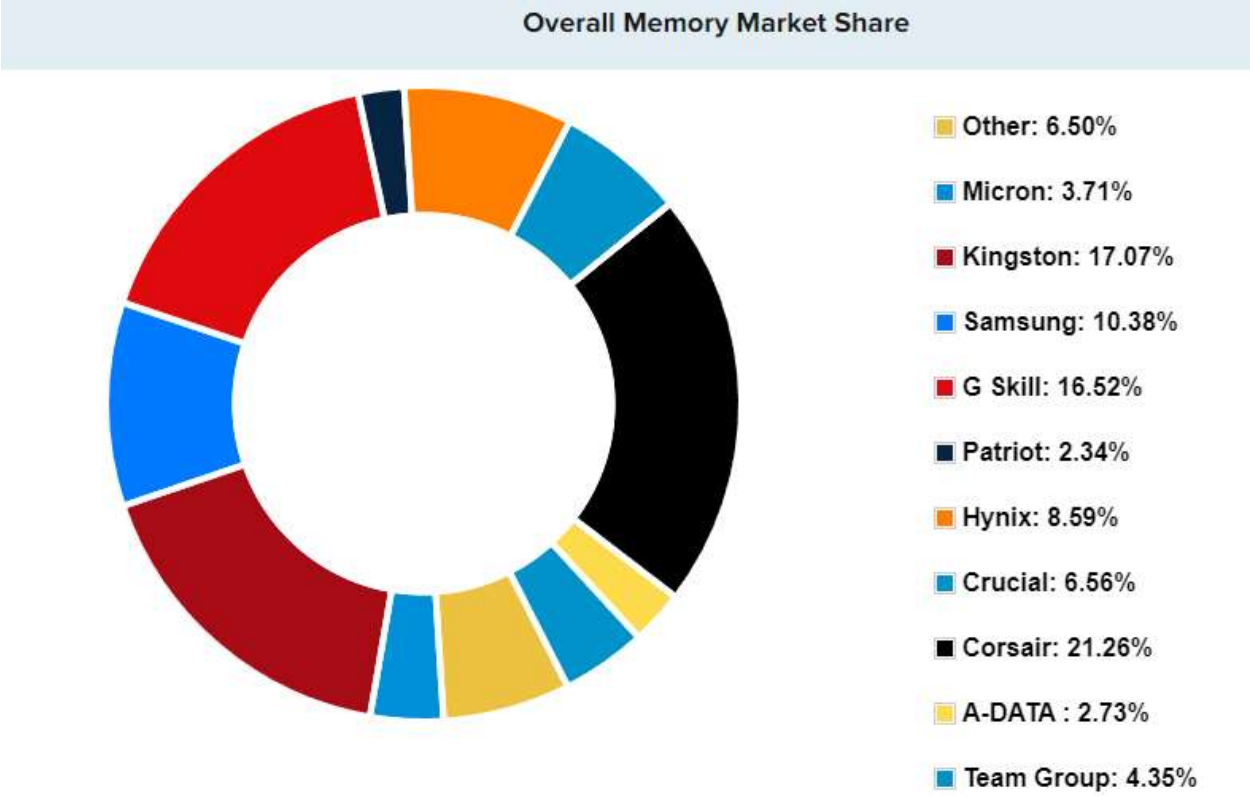


Fig. 8 – Fabricantes de memoria RAM

Fuente: https://www.memorybenchmark.net/30dayshare.html

Fig. 9 – Top ten de celulares con Android

Fuente: https://www.androidbenchmark.net/passmark_chart.html

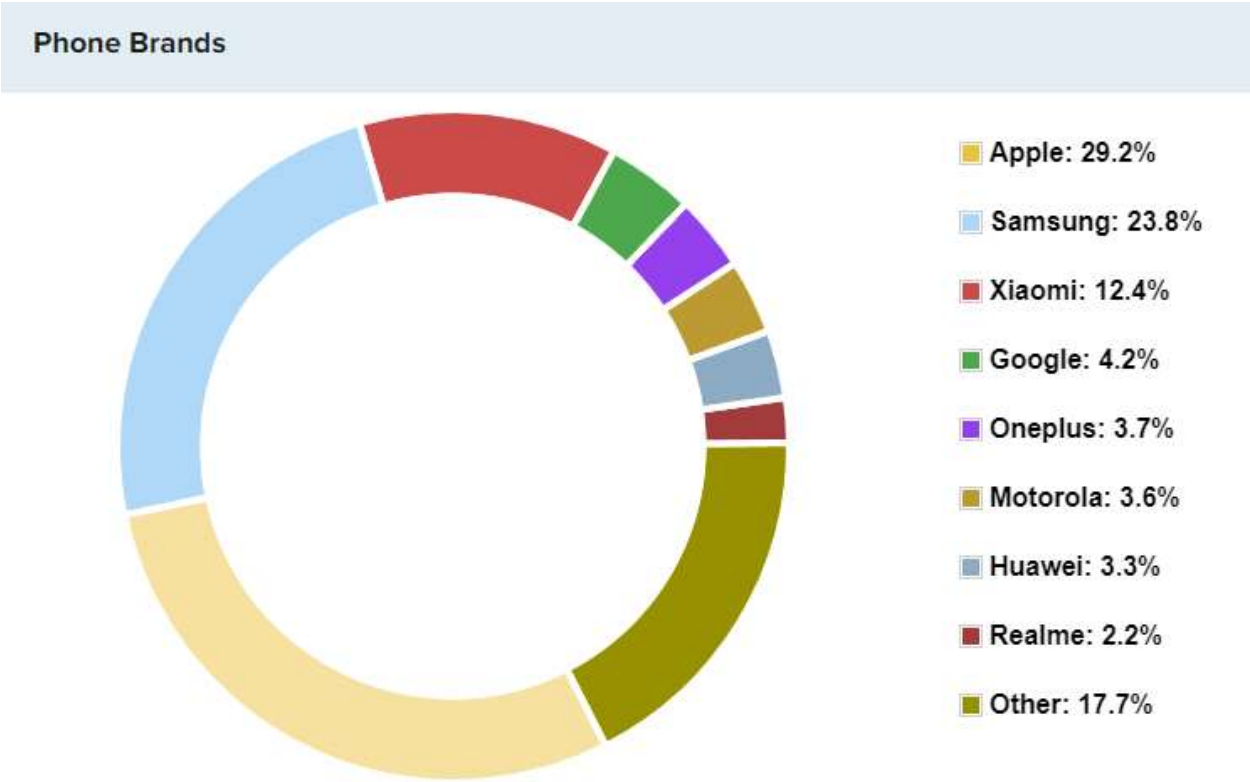


Fig. 10 – Fabricantes de celulares con Android

Fuente: https://www.androidbenchmark.net/30dayshare.html

Rank	System	Cores	(PFlop/s)	(PFlop/s)	(kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107
6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404
7	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
8	**MareNostrum 5 ACC** - BullSequana XH3000, Xeon Platinum 8460Y+ 40C 2.3GHz, NVIDIA H100 64GB, Infiniband NDR200, EVIDEN EuroHPC/BSC Spain	680,960	138.20	265.57	2,560
9	**Eos NVIDIA DGX SuperPOD** - NVIDIA DGX H100, Xeon Platinum 8480C 56C 3.8GHz, NVIDIA H100, Infiniband NDR400, Nvidia NVIDIA Corporation United States	485,888	121.40	188.65	
10	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438

Fig. 11 – Top ten de supercomputadoras (Nov 2023)

Fuente: https://www.top500.org/lists/top500/list/2023/11/

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	HPCG (TFlop/s)
1	4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	16004.50
2	1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	14054.00
3	5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	4586.95
4	6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	238.70	3113.94
5	7	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	2925.75
6	12	**Perlmutter** - HPE Cray EX 235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-11, HPE DOE/SC/LBNL/NERSC United States	888,832	79.23	1905.00
7	10	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	1795.67
8	13	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	1622.51
9	18	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, EVIDEN Forschungszentrum Juelich (FZJ) Germany	449,280	44.12	1275.36
10	50	**AOBA-S** - SX-Aurora TSUBASA B401-8, Vector Engine Type 30A 16C 1.6GHz, Infiniband NDR 200, NEC Cyberscience Center, Tohoku University Japan	64,512	17.22	1089.00

Fig. 12 – Top ten de supercomputadoras (HPCG) (Nov 2023)

Fuente: https://www.top500.org/lists/hpcg/2023/11/

APÉNDICE G – LOS MICROPROCESADORES 80x86

G.1 EL 8086

G.1.1 Estructura interna

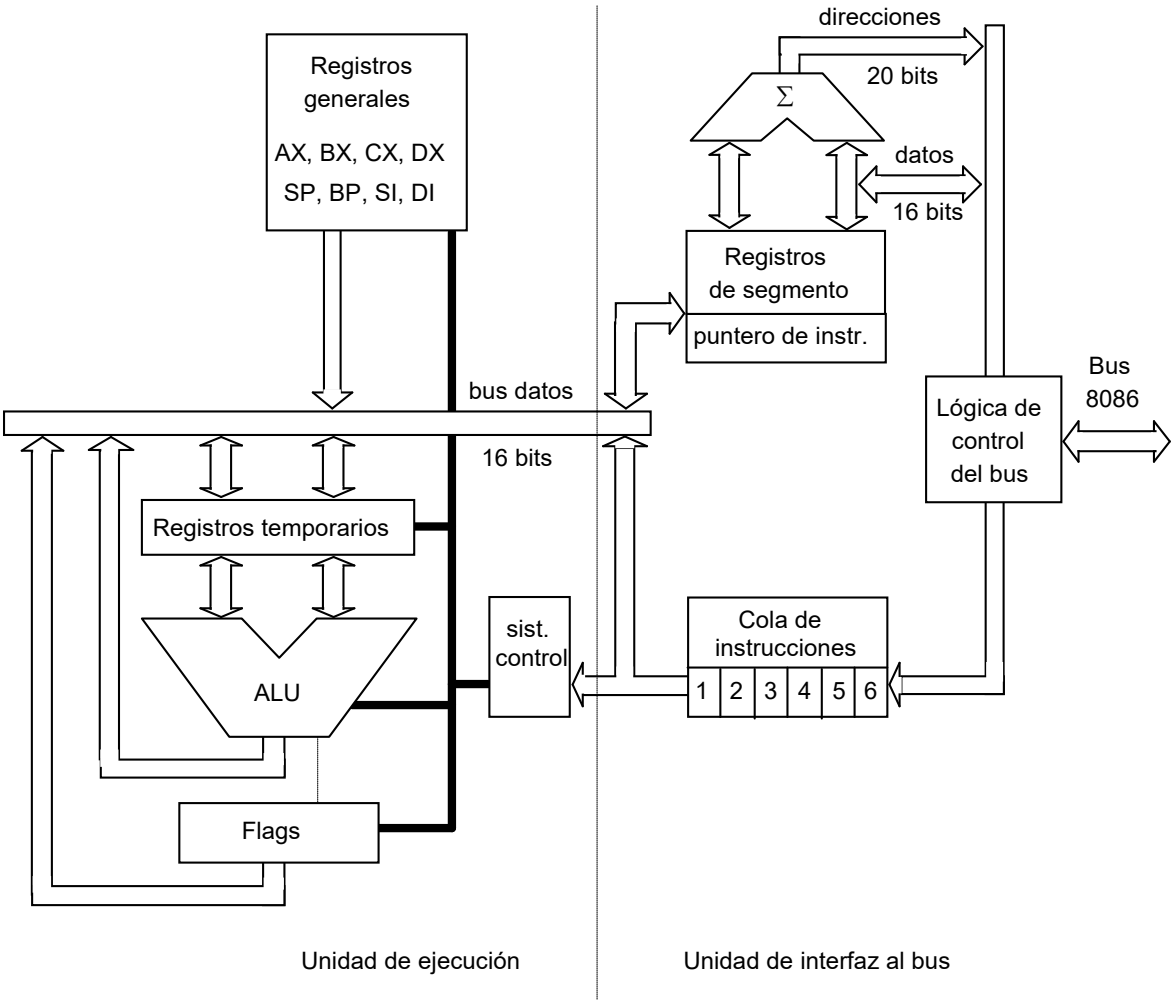


Figura G.1.1 - Diagrama funcional simplificado del 8086

Para el microprocesador Intel 8088, el diagrama es el mismo, con la diferencia que la cola de instrucciones es de 4 bytes en lugar de 6 y que el bus de datos externo es de 8 bits en lugar de 16.

G.1.2 Listado de conexiones

Esta lista aplica tanto al modo mínimo como al modo máximo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
AD15-AD0	2-16, 39	I/O	ADDRESS DATA BUS: These lines constitute the time multiplexed memory/IO address (T1) and data (T2, T3, TW, T4) bus. A0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D7-D0. It is LOW during Ti when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use A0 to condition chip select functions (See $\overline{BHE}$). These lines are active HIGH and are held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence".
A19/S6 A18/S5 A17/S4 A16/S3	35-38	O	ADDRESS/STATUS: During T1, these are the four most significant address lines for memory operations. During I/O operations these lines are LOW. During memory and I/O operations, status information is available on these lines during T2, T3, TW, T4. S6 is always LOW. The status of the interrupt enable FLAG bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines are held at high impedance to the last valid logic level during local bus "hold acknowledge" or "grant sequence". S4 / S3 / CHARACTERISTICS 0 / 0 / Alternate Data 0 / 1 / Stack 1 / 0 / Code or None 1 / 1 / Data
$\overline{BHE}$/S7	34	O	BUS HIGH ENABLE/STATUS: During T1 the bus high enable signal ($\overline{BHE}$) should be used to enable data onto the most significant half of the data bus, pins D15-D8. Eight bit oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S7 status information is available during T2, T3 and T4. The signal is active LOW, and is held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence", it is LOW during T1 for the first interrupt acknowledge cycle. BHE / A0 / CHARACTERISTICS 0 / 0 / Whole Word 0 / 1 / Upper Byte From/to Odd Address 1 / 0 / Lower Byte From/to Even address 1 / 1 / None
$\overline{RD}$	32	O	READ: Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the M/$\overline{IO}$ or $\overline{S2}$ pin. This signal is used to read devices which reside on the 80C86 local bus. $\overline{RD}$ is active LOW during T2, T3 and TW of any read cycle, and is guaranteed to remain HIGH in T2 until the 80C86 local bus has floated. This line is held at a high impedance logic one state during "hold acknowledge" or "grand sequence".
READY	22	I	READY: is the acknowledgment from the addressed memory or I/O device that will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 82C84A Clock Generator to form READY. This signal is active HIGH. The 80C86 READY input is not synchronized. Correct operation is not guaranteed if the Setup and Hold Times are not met.

Tabla G.1.1a - Listado de conexiones en el 8086.

Esta lista aplica tanto al modo mínimo como al modo máximo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
INTR	18	I	INTERRUPT REQUEST: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
$\overline{TEST}$	23	I	TEST: input is examined by the "Wait" instruction. If the $\overline{TEST}$ input is LOW execution continues, otherwise the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
NMI	17	I	NON-MASKABLE INTERRUPT: is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
RESET	21	I	RESET: causes the processor to immediately terminate its present activity. The signal must transition LOW to HIGH and remain active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	CLOCK: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
VCC	40		VCC: +5V power supply pin. A 0.1µF capacitor between pins 20 and 40 is recommended for decoupling.
GND	1, 20		GND: Ground. Note: both must be connected. A 0.1µF capacitor between pins 1 and 20 is recommended for decoupling.
MN/$\overline{MX}$	33	I	MINIMUM/MAXIMUM: Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

Tabla G.1.1b Listado de conexiones en el 8086.

Esta lista aplica sólo al modo mínimo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
M/$\overline{IO}$	28	O	STATUS LINE: logically equivalent to $\overline{S2}$ in the maximum mode. It is used to distinguish a memory access from an I/O access. M/$\overline{IO}$ becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (M = HIGH, I/O = LOW). M/$\overline{IO}$ is held to a high impedance logic one during local bus "hold acknowledge".
$\overline{WR}$	29	O	WRITE: indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the M/$\overline{IO}$ signal. $\overline{WR}$ is active for T2, T3 and TW of any write cycle. It is active LOW, and is held to high impedance logic one during local bus "hold acknowledge".
$\overline{INTA}$	24	O	INTERRUPT ACKNOWLEDGE: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3 and TW of each interrupt acknowledge cycle. Note that $\overline{INTA}$ is never floated.
ALE	25	O	ADDRESS LATCH ENABLE: is provided by the processor to latch the address into the 82C82/82C83 address latch. It is a HIGH pulse active during clock LOW of T1 of any bus cycle. Note that ALE is never floated.

Tabla G.1.1c - Listado de conexiones en el 8086.

Esta lista aplica sólo al modo mínimo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
DT/$\overline{R}$	27	O	DATA TRANSMIT/RECEIVE: is needed in a minimum system that desires to use a data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/$\overline{R}$ is equivalent to $\overline{S1}$ in maximum mode, and its timing is the same as for M/$\overline{IO}$ (T = HIGH, R = LOW). DT/$\overline{R}$ is held to a high impedance logic one during local bus "hold acknowledge".
$\overline{DEN}$	26	O	DATA ENABLE: provided as an output enable for a bus transceiver in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle it is active from the middle of T2 until the middle of T4, while for a write cycle it is active from the beginning of T2 until the middle of T4. $\overline{DEN}$ is held to a high impedance logic one during local bus "hold acknowledge".
HOLD HLDA	31, 30	I O	HOLD: indicates that another master is requesting a local bus "hold". To be an acknowledged, HOLD must be active HIGH. The processor receiving the "hold" will issue a "hold acknowledge" (HLDA) in the middle of a T4 or TI clock cycle. Simultaneously with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will lower HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. HOLD is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the setup time.

Tabla G.1.1d - Listado de conexiones en el 8086.

Esta lista aplica sólo al modo máximo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
$\overline{S0}$ $\overline{S1}$ $\overline{S2}$	26 27 28	O O O	STATUS: is active during T4, T1 and T2 and is returned to the passive state (1, 1, 1) during T3 or during TW when READY is HIGH. This status is used by the 82C88 Bus Controller to generate all memory and I/O access control signals. Any change by $\overline{S2}$, $\overline{S1}$ or $\overline{S0}$ during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 or TW is used to indicate the end of a bus cycle. These signals are held at a high impedance logic one state during "grant sequence".

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	CHARACTERISTICS
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

Tabla G.1.1e - Listado de conexiones en el 8086.

Esta lista aplica sólo al modo máximo.

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
$\overline{RQ}/\overline{GT0}$ $\overline{RQ}/\overline{GT1}$	31, 30	I/O	REQUEST/GRANT: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT0}$ having higher priority than $\overline{RQ}/\overline{GT1}$. $\overline{RQ}/\overline{GT}$ has an internal pull-up bus hold device so it may be left unconnected. The request/grant sequence is as follows (see $\overline{RQ}/\overline{GT}$ Sequence Timing) 1. A pulse of 1 CLK wide from another local bus master indicates a local bus request ("hold") to the 80C86 (pulse 1). 2. During a T4 or TI clock cycle, a pulse 1 CLK wide from the 80C86 to the requesting master (pulse 2) indicates that the 80C86 has allowed the local bus to float and that it will enter the "grant sequence" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "grant sequence". 3. A pulse 1 CLK wide from the requesting master indicates to the 80C86 (pulse 3) that the "hold" request is about to end and that the 80C86 can reclaim the local bus at the next CLK. The CPU then enters T4 (or TI if no bus cycles pending). Each Master-Master exchange of the local bus is a sequence of 3 pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active low. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conditions are met: 1. Request occurs on or before T2. 2. Current cycle is not the low byte of a word (on an odd address). 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made the two possible events will follow: 1. Local bus will be released during the next cycle. 2. A memory cycle will start within three clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
$\overline{LOCK}$	29	O	LOCK: output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and is held at a high impedance logic one state during "grant sequence". In MAX mode, $\overline{LOCK}$ is automatically generated during T2 of the first $\overline{INTA}$ cycle and removed during T2 of the second $\overline{INTA}$ cycle.
QS1, QSO	24, 25	O	QUEUE STATUS: The queue status is valid during the CLK cycle after which the queue operation is performed. QS1 and QS0 provide status to allow external tracking of the internal 80C86 instruction queue. Note that QS1, QS0 never become high impedance. QSI / QSO: 0 0 No Operation 0 1 First byte of op code from queue 1 0 Empty the queue 1 1 Subsequent byte from queue

Tabla G.1.1f - Listado de conexiones en el 8086.

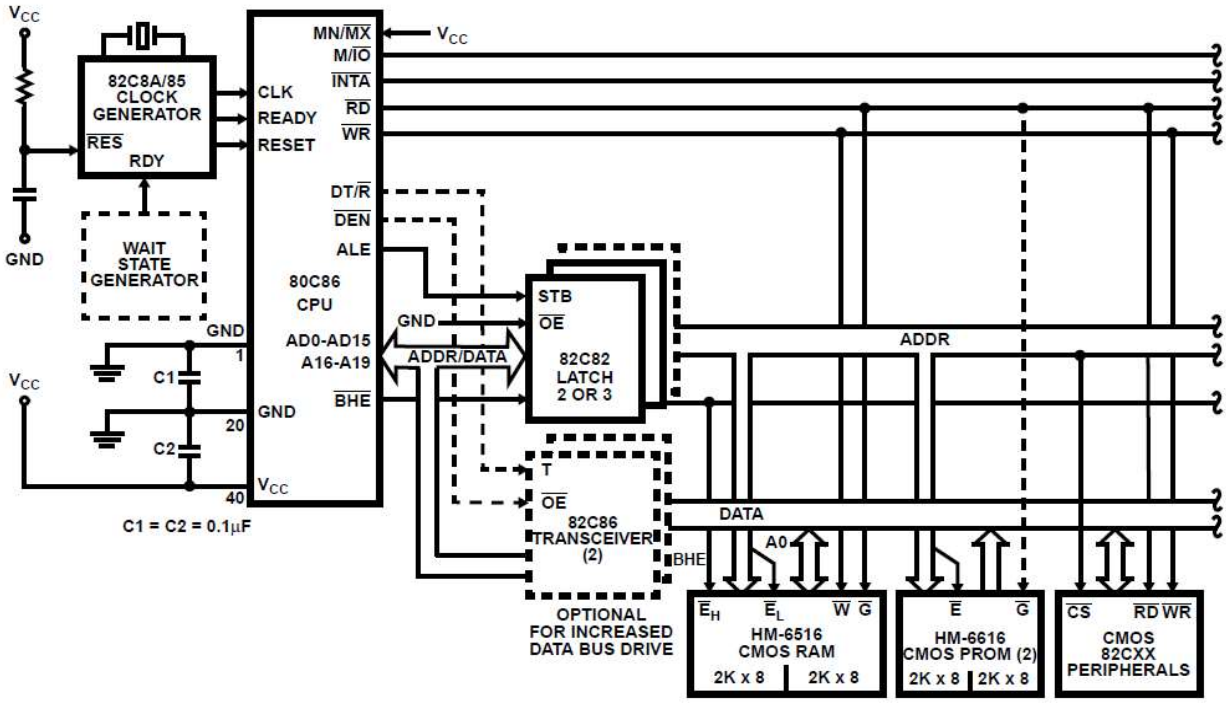


Figura G.1.2a – Circuito de aplicación del 8086 en modo mínimo

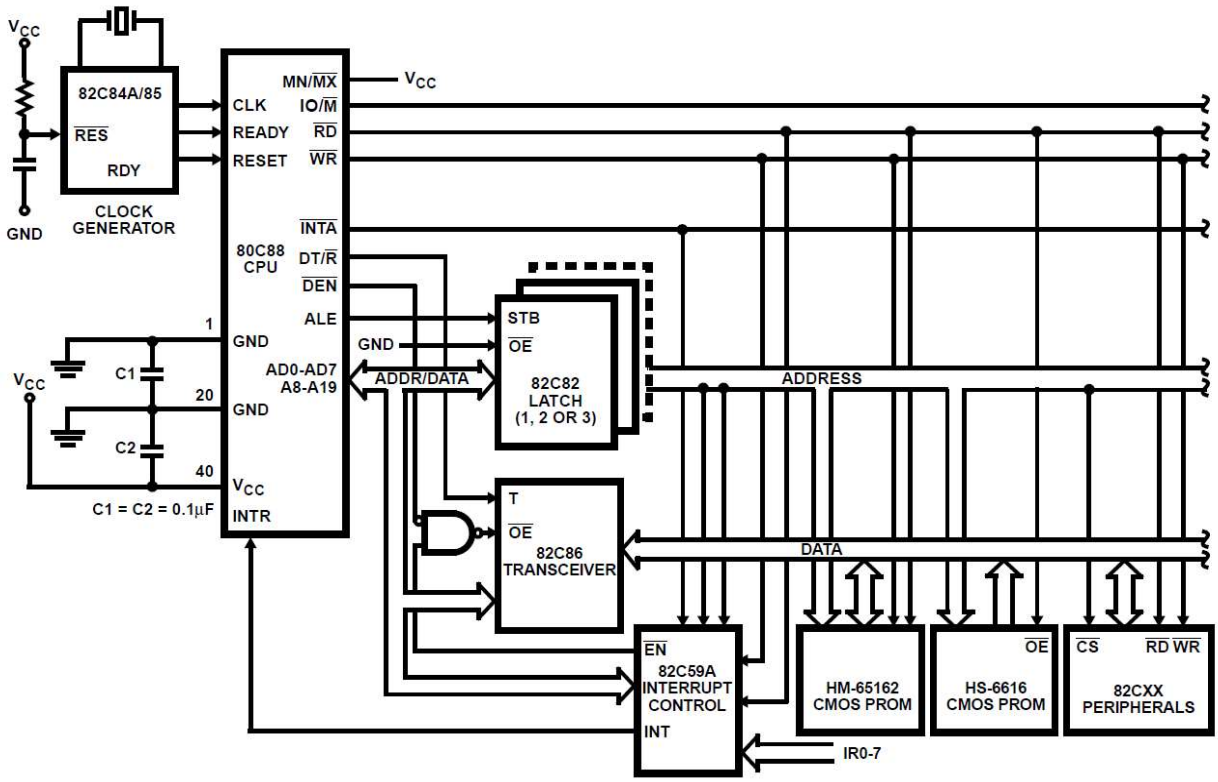


Figura G.1.2b – Circuito de aplicación del 8088 en modo mínimo

G.1.3 Modelo programable

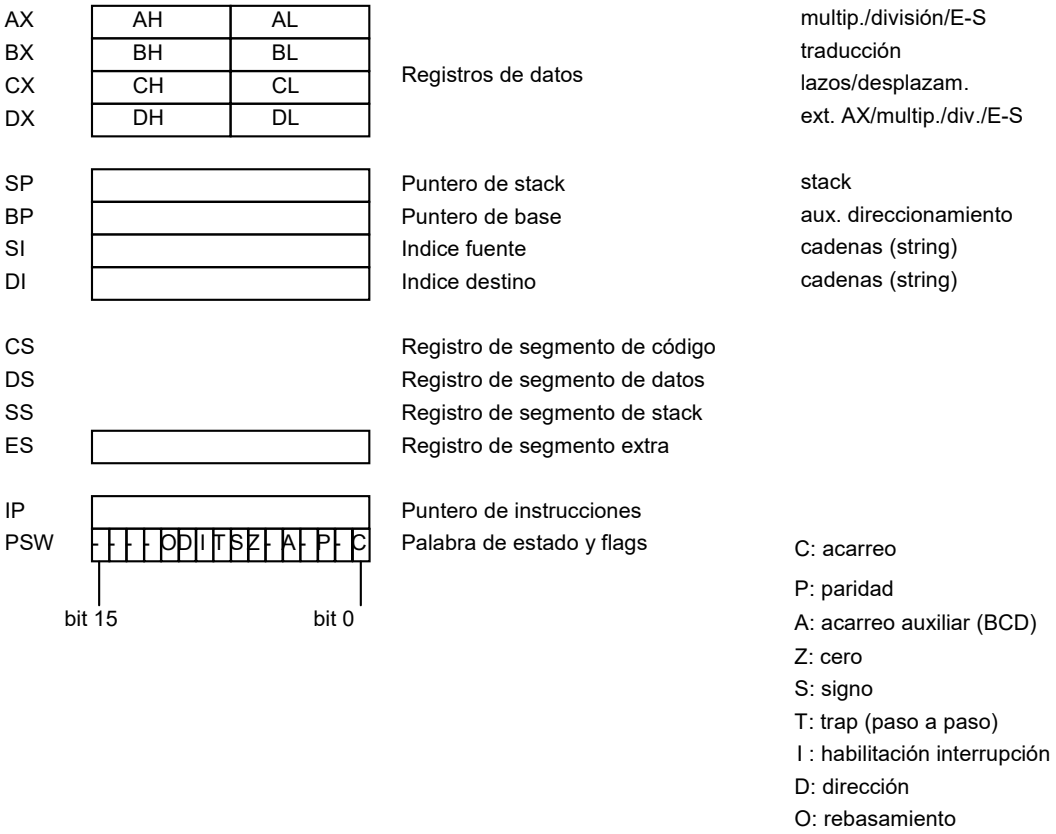


Figura G.1.3 - Registros del 8086/8088

G.1.4 Organización de la memoria

En los sistemas con 8086 la memoria está organizada por bytes, es decir que a cada dirección de memoria le corresponde un byte. Sin embargo, en las operaciones de lectura (memoria a CPU) se transfieren siempre dos bytes en el 8086, aprovechando su bus de datos de 16 bits y sólo uno en el 8088. El siguiente es un ejemplo para 8086:

MOV AX, (0724H) ;mueve el contenido de 0724H al registro AX, se codifica: A1 24 07

Dirección	Dato
725	55
724	02

AX= 5502H AH=55
AL=02

Esta operación se realiza en un ciclo de bus, por estar el dato en una dirección par. En este otro ejemplo, también para 8086, se hace la misma operación, pero desde una locación impar:

MOV AX, (0725H) ;mueve el contenido de 0725H al registro AX, se codifica: A1 25 07

Dirección	Dato
726	55
725	02

AX= 5502H AH=55
AL=02

En este caso, se obtiene el mismo resultado, pero la operación se realiza en dos ciclos de bus y por lo tanto demora mas. Esta situación puede evitarse colocando los datos en direcciones pares.

G.1.5 Modos de direccionamiento

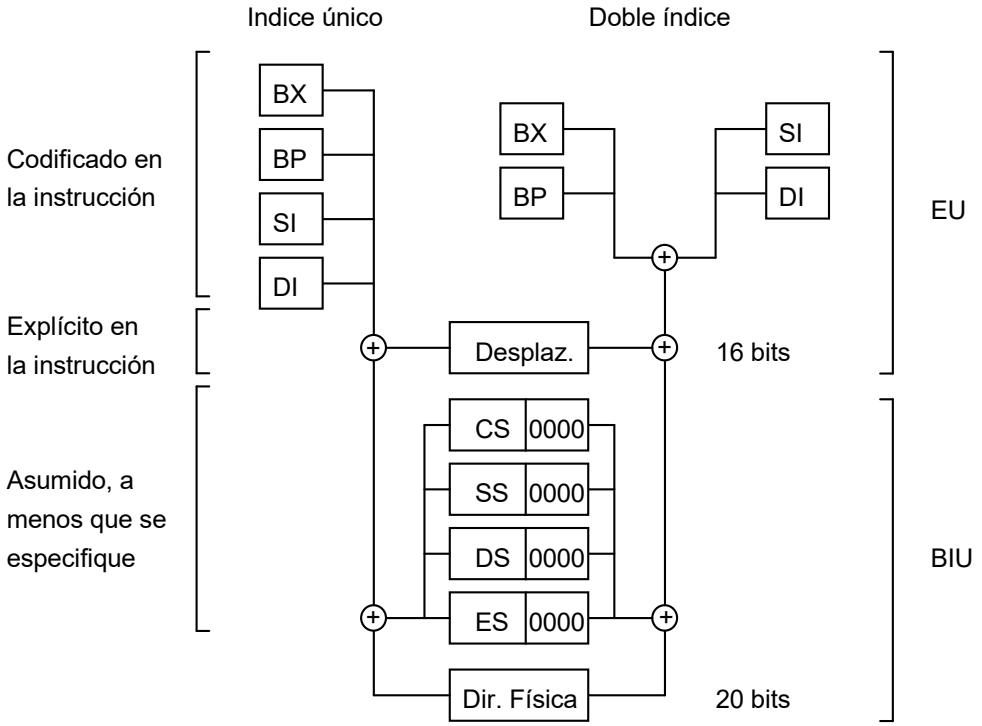


Figura G.1.4 – Cálculo de la dirección física

Operando inmediato o registro

MOV AX, 2CH

MOV AX,BX

; no hay ciclos de bus

Directo

MOV AX,DIR ; DIR es una dirección

Indirecto via registro

MOV AX,[BX] ; ver índice único en la figura

Relativo a registro base

MOV AX, [BX+14H] ; con BX se usa DS, con BP se usa SS

Relativo a registro índice (indexado)

MOV AX,TABLA[SI] ; ejemplo: lista de datos

Base e índice

MOV AX, [BX][SI] ; ejemplo: varias listas de datos

Relativo con base e índice

MOV AX, TABLA[BX][SI] ; ejemplo: matriz en un stack

Direccionamiento de saltos

Relativo

JNZ ABAJO ; 8 bits ... EBA = [IP]+Desp

Directo intrasegmento

JE ABAJO ; 8 o 16 bits ... EBA = ([IP]+Desp) + CS

Indirecto intrasegmento

JMP [BX] Desp ; EBA = EA

Directo intersegmento

JMP FARLABEL ; CS:IP ... el destino se definió como FAR

Indirecto intersegmento

JMP TABLA[BX]

EBA: Effective Branch Address (Dirección Efectiva de Bifurcación)

CS:IP: Par de registros que sumados dan el PC de 20 bits

G.1.6 El sistema de interrupciones

Dirección	Contenido	Vector	Descripción
3FE	CS 255	Vector 255	Disponible
3FC	IP 255	"	Usuario
..	..	..	..
82	CS 32	Vector 32	"
80	IP 32	"	"
7E	CS 31	Vector 31	Reservado
7C	IP 31	"	INTEL
..	..	..	..
12	CS 4	Vector 4	Overflow
10	IP 4	"	"
E	CS 3	Vector 3	Break-point
C	IP 3	"	"
A	CS 2	Vector 2	NMI
8	IP 2	"	"
6	CS 1	Vector 1	Single step
4	IP 1	"	"
2	CS 0	Vector 0	Divide error
0	IP 0	"	"

Tabla G.1.2 Asignación de vectores en el 8086

Interrupciones de Hardware

NMI: Interrupción no-enmascarable - Se activa por flanco en el pin 17, causando una interrupción de tipo 2.

INTR: Requerimiento de interrupción (enmascarable) - Se activa por nivel en el pin 18 y es muestreada en el último ciclo de reloj de cada instrucción para determinar si el procesador deberá ejecutar una operación de reconocimiento de interrupción.

Producida la condición de interrupción, se salvan en el stack el CS, IP, FLAGS y se transfiere el control a la rutina de atención de interrupción indicada por CS:IP, tomados como muestra la tabla G.1.2. Generalmente se controla con el controlador de interrupciones Intel 8259.

Interrupciones de Software

TIPO 0 - Error de división - Ocurre en una división cuando el cociente excede el valor admitido por el registro que debe almacenarlo (AX o AL). Es no-enmascarable y utiliza el vector cero. Ejemplo: división por cero.

TIPO 1 - Paso a paso - Ocurre una instrucción después que se haya fijado el bit T (trap) en el registro de flags. La operación se realiza salvando los flags en el stack, luego alterando la copia del bit T en el stack y recuperando luego los flags del stack, que vuelven con el bit T fijado. Ejemplo: para hacer trace en el DEBUG y ejecutar instrucción por instrucción.

TIPO 3 - Instrucción INT3 - Es una interrupción de un byte que utiliza el vector 3. Ejemplo: para instalar break-points el DEBUG reemplaza la posición deseada por INT3.

TIPO 4 - Instrucción INTO - Ocurre cuando se ejecuta la instrucción INTO y el flag de Overflow está fijado. Permite analizar la condición de rebasamiento por una subrutina. Usa el vector 4.

TIPO n - Instrucción INTn, n≠3 - Interrupción de usuario de dos bytes. El primero es el código de operación y el segundo el número de interrupción n. Ejemplo: llamadas a servicios del sistema operativo (video, archivos, etc.). Puede usarse para simular cualquier interrupción, inclusive las de hardware.

G.1.7 Acceso directo a memoria

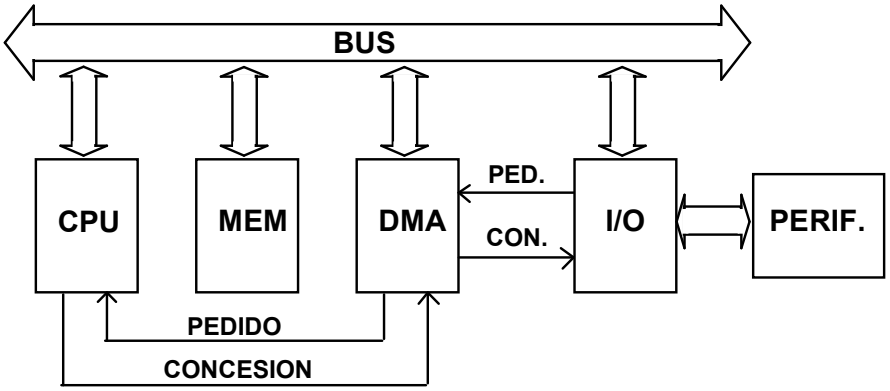


Figura G.1.5 - Configuración circuital

El 8086 soporta protocolos para transferir el control del bus local entre el mismo y otros dispositivos capaces de actuar como maestros del bus. La configuración en modo mínimo ofrece un nivel de intercambio (handshake) de señales similar a los sistemas 8080 y 8085. El modo máximo provee un protocolo mejorado de secuencia de pulsos diseñado para optimizar la utilización de las patitas de la CPU y a la vez extiende las configuraciones del sistema a dos niveles priorizados de maestros del bus.

Estos protocolos son simplemente técnicas para arbitrar el control del bus local de la CPU y no deberían ser confundidas con la necesidad de arbitraje del bus del sistema.

Modo mínimo (HOLD/HLDA)

El sistema 8086 en modo mínimo usa una entrada de petición (HOLD) a la CPU y una salida de concesión (HLDA) de la CPU (figura G.1.5). Para ganar el control del bus, un dispositivo debe poner HOLD a la CPU y esperar por HLDA antes de manejar el bus. Cuando el 8086 puede ceder el bus, flota las líneas de control RD*, WR*, INTA*, M/IO*, DEN* y DT/R* y las líneas multiplexadas de direcciones/datos/estados. La señal ALE no se pone en tri-state.

La CPU reconoce el pedido con HLDA para permitir al solicitante tomar el control del bus. El solicitante debe mantener el pedido de HOLD activo hasta tanto no necesite mas del bus. El pedido de HOLD al 8086 afecta directamente a la unidad de ejecución. La CPU continuará ejecutando desde su cola interna hasta que sean necesarias más instrucciones o se requiera una transferencia de operando. Esto permite un alto grado de solapamiento entre la CPU y el maestro auxiliar del bus.

Cuando el solicitante retira la señal HOLD, el 8086 responderá retirando la señal HLDA. La CPU no manejará nuevamente el bus ni las señales de control, que permanecerán en tri-state hasta que necesite realizar una transferencia en el bus. De esta forma, el 8086 puede aún estar ejecutando desde su cola interna cuando el dispositivo que pidió HOLD está manejando el bus. Para prevenir que las líneas de comando deriven por debajo del nivel V_{IH} durante las transiciones de control del bus, deberían conectarse resistores de pull-up de 22 KΩ a las líneas de comando (RD*, WR*, INTA* y M/IO*).

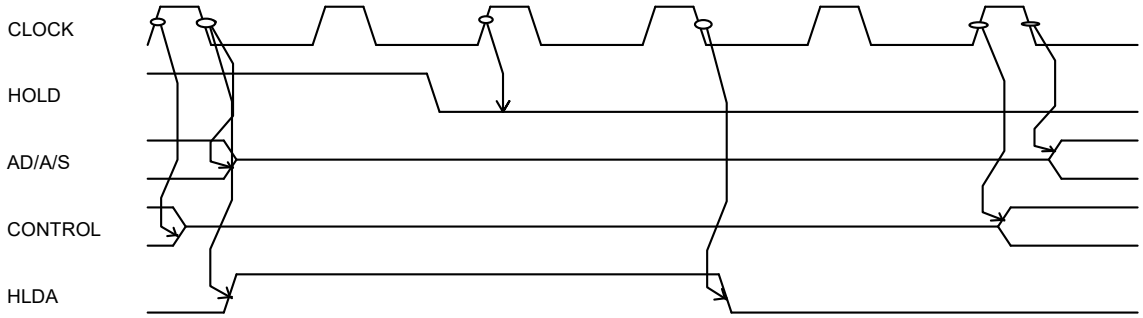


Figura G.1.6 - Diagrama de tiempos de la secuencia hold/hlda

Configuración DMA en modo mínimo (ejemplo usando 8237)

Un uso típico de las señales HOLD/HLDA en el modo mínimo en un sistema 8088 es el intercambio del control del bus con dispositivos DMA tales como el 8237, que es un controlador de DMA de Intel. La figura G.1.7 ilustra la interconexión para este tipo de configuración, usando 8237.

La configuración del 8086 en modo máximo soporta un protocolo significativamente diferente para la transferencia del control del bus. Cuando es visto en comparación con la secuencia HOLD/HLDA del modo mínimo, el protocolo aparece como difícil de implementar externamente. De todos modos, es necesario entender la intención del protocolo y su propósito dentro de la arquitectura del sistema.

La secuencia RQ*/GT* del modo máximo pretende transferir el control del bus local de la CPU entre la CPU y maestros del bus alternativos los cuales residen totalmente en el bus local y comparten completamente la interfaz de la CPU al bus del sistema. La interfaz incluye los latch de direcciones, los transceivers de datos, el controlador del bus 8288 y el árbitro del bus 8289.

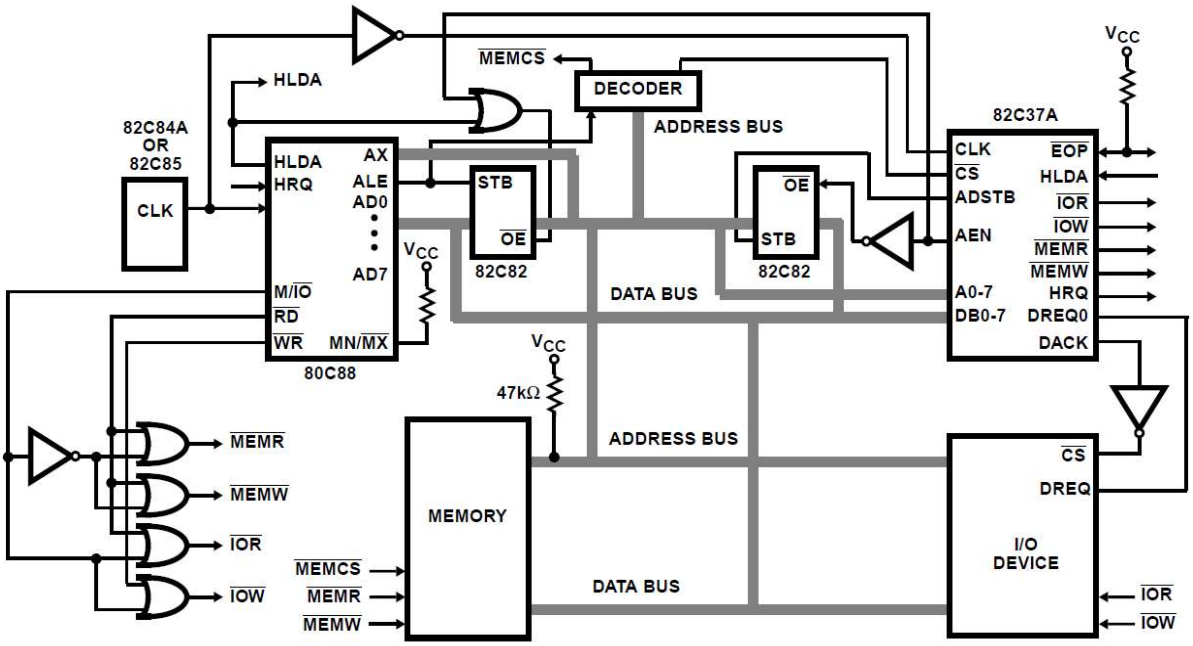


Figura G.1.7 - DMA usando el 8237

El 8086 en Modo máximo - Utilización de RQ/GT**

El protocolo RQ*/GT* fue desarrollado para permitir que hasta dos procesadores de extensión del conjunto de instrucciones (coprocesadores) u otros procesadores especiales (como el procesador de E/S 8089 en modo local) residan directamente en el bus local de la CPU. Cada patilla de RQ*/GT* del 8086 soporta el protocolo completo para el intercambio del control del bus.

La secuencia consiste en un pedido (ReQuest) desde el maestro del bus alternativo para ganar el control del bus, una concesión (GranT) desde la CPU para indicar que el bus ha sido abandonado y un pulso de liberación desde el maestro alternativo cuando concluye. Las dos patitas de RQ*/GT* (RQ*/GT0* y RQ*/GT1*) están priorizadas, teniendo RQ*/GT0* la máxima prioridad. La priorización sólo ocurre cuando ambos pedidos son recibidos antes de que uno de ellos sea concedido.

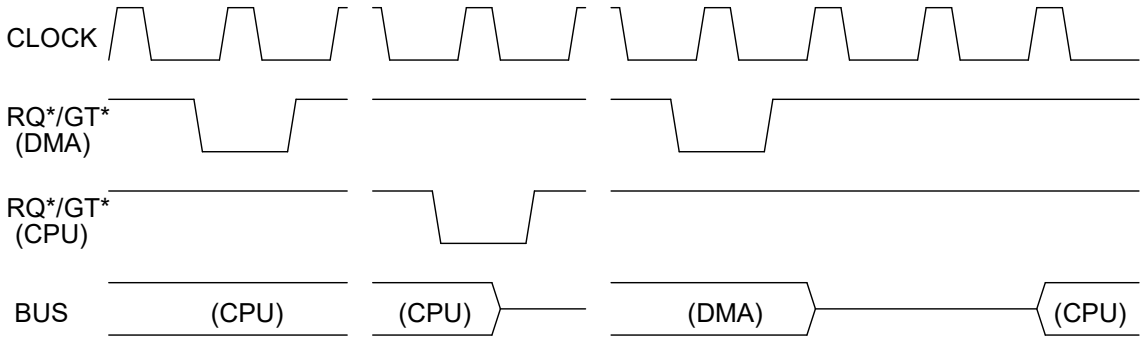


Figura G.1.8 - Diagrama de tiempos de la secuencia RQ*/GT*

La interacción de la secuencia pedido/concesión con la unidad de interfaz del bus es similar a HOLD/HLDA. La CPU continúa ejecutando hasta tanto requiera una transferencia en el bus para obtener instrucciones adicionales o datos.

Ante la recepción de un pulso de solicitud del bus, el 8086 flota los buses multiplexados de direcciones, datos y estados, las líneas de estado S0*, S1* y S2*, la patilla LOCK y RD*. Esta acción no inhabilita que las salidas de comando del 8288 manejen el bus de control y no inhabilita que los latch de direcciones manejen el bus de direcciones. El 8288 contiene resistores a Vcc internos en las líneas de estado S0*, S1* y S2* para mantener el estado pasivo mientras las salidas del 8086 están en estado de alta impedancia. El estado pasivo previene que el 8288 inicie cualquier comando o active DEN para habilitar los transceivers del bus de datos. Si el dispositivo que envía RQ* no usa el 8288, debe inhabilitar las salidas de comando del 8288 mediante la inhabilitación de la entrada AEN*. Asimismo, los latch de direcciones que no son usados por el dispositivo solicitante deben estar inhabilitados.

G.2 EL 80386

G.2.1 Estructura interna

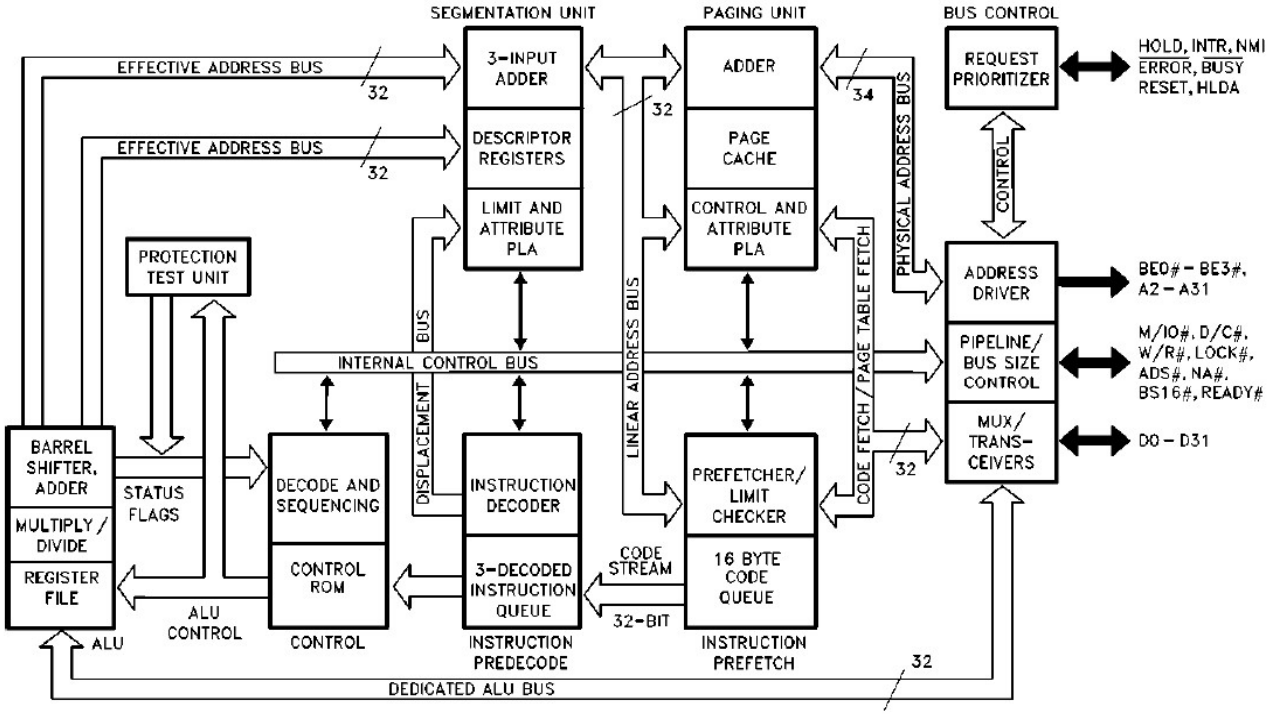


Figura G.2.1 - Diagrama funcional simplificado del 80386

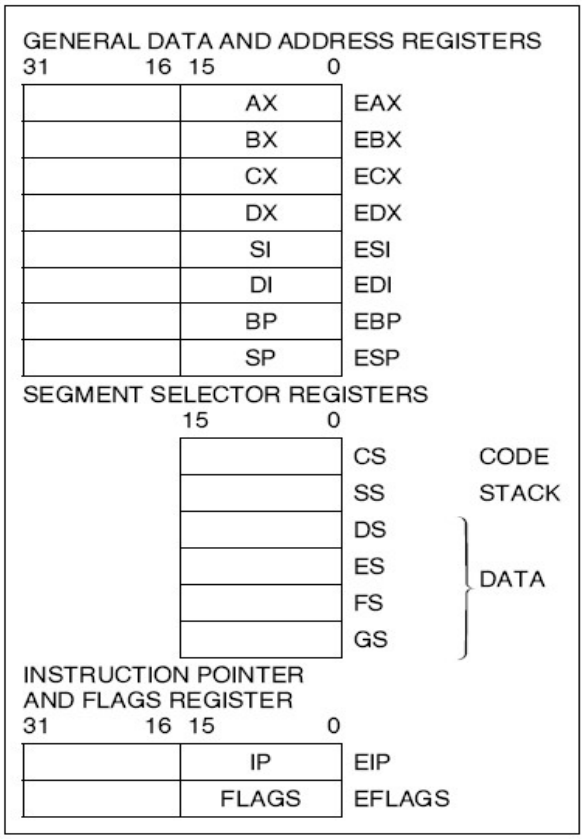
G.2.2 Listado de conexiones

Symbol	Type	Name and Function
CLK2	I	**CLK2** provides the fundamental timing for the Intel386 DX.
$D_{31}-D_0$	I/O	**DATA BUS** inputs data during memory, I/O and interrupt acknowledge read cycles and outputs data during memory and I/O write cycles.
$A_{31}-A_2$	O	**ADDRESS BUS** outputs physical memory or port I/O addresses.
BE0#–BE3#	O	**BYTE ENABLES** indicate which data bytes of the data bus take part in a bus cycle.
W/R#	O	**WRITE/READ** is a bus cycle definition pin that distinguishes write cycles from read cycles.
D/C#	O	**DATA/CONTROL** is a bus cycle definition pin that distinguishes data cycles, either memory or I/O, from control cycles which are: interrupt acknowledge, halt, and instruction fetching.
M/IO#	O	**MEMORY I/O** is a bus cycle definition pin that distinguishes memory cycles from input/output cycles.
LOCK#	O	**BUS LOCK** is a bus cycle definition pin that indicates that other system bus masters are denied access to the system bus while it is active.
ADS#	O	**ADDRESS STATUS** indicates that a valid bus cycle definition and address (W/R#, D/C#, M/IO#, BE0#, BE1#, BE2#, BE3# and $A_{31}-A_2$) are being driven at the Intel386 DX pins.
NA#	I	**NEXT ADDRESS** is used to request address pipelining.
READY#	I	**BUS READY** terminates the bus cycle.
BS16#	I	**BUS SIZE 16** input allows direct connection of 32-bit and 16-bit data buses.
HOLD	I	**BUS HOLD REQUEST** input allows another bus master to request control of the local bus.

Symbol	Type	Name and Function
HLDA	O	**BUS HOLD ACKNOWLEDGE** output indicates that the Intel386 DX has surrendered control of its local bus to another bus master.
BUSY#	I	**BUSY** signals a busy condition from a processor extension.
ERROR#	I	**ERROR** signals an error condition from a processor extension.
PEREQ	I	**PROCESSOR EXTENSION REQUEST** indicates that the processor extension has data to be transferred by the Intel386 DX.
INTR	I	**INTERRUPT REQUEST** is a maskable input that signals the Intel386 DX to suspend execution of the current program and execute an interrupt acknowledge function.
NMI	I	**NON-MASKABLE INTERRUPT REQUEST** is a non-maskable input that signals the Intel386 DX to suspend execution of the current program and execute an interrupt acknowledge function.
RESET	I	**RESET** suspends any operation in progress and places the Intel386 DX in a known reset state. See **Interrupt Signals** for additional information.
N/C	—	**NO CONNECT** should always remain unconnected. Connection of a N/C pin may cause the processor to malfunction or be incompatible with future steppings of the Intel386 DX.
V_{CC}	I	**SYSTEM POWER** provides the +5V nominal D.C. supply input.
V_{SS}	I	**SYSTEM GROUND** provides 0V connection from which all inputs and outputs are measured.

Tabla G.2.1 - Listado de conexiones en el 80386

G.2.3 Modelo programable



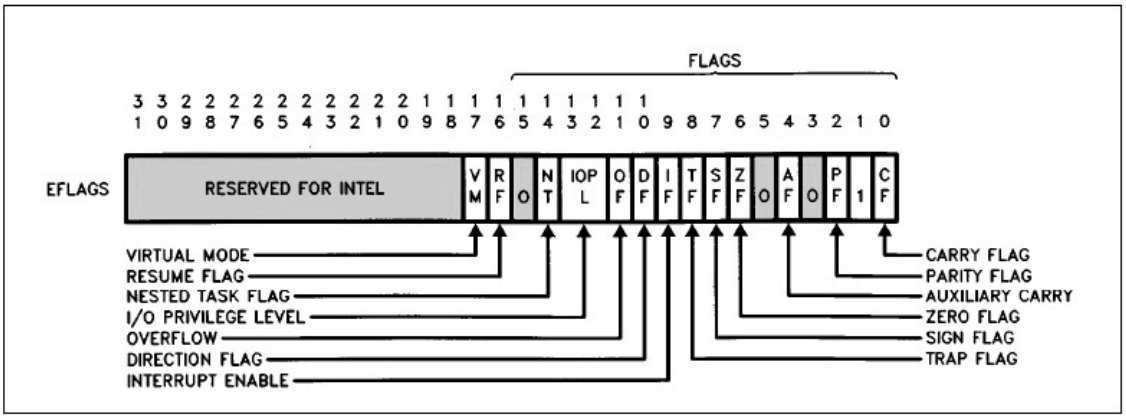


Fig. G.2.2 – Registros del 80386

G.2.4 Organización de la memoria

En los sistemas con 80386 la memoria también está organizada por bytes, es decir que a cada dirección de memoria le corresponde un byte. Sin embargo, en las operaciones de lectura (memoria a CPU) se transfieren siempre cuatro bytes, aprovechando su bus de datos de 32 bits.

G.2.5 Modos de direccionamiento

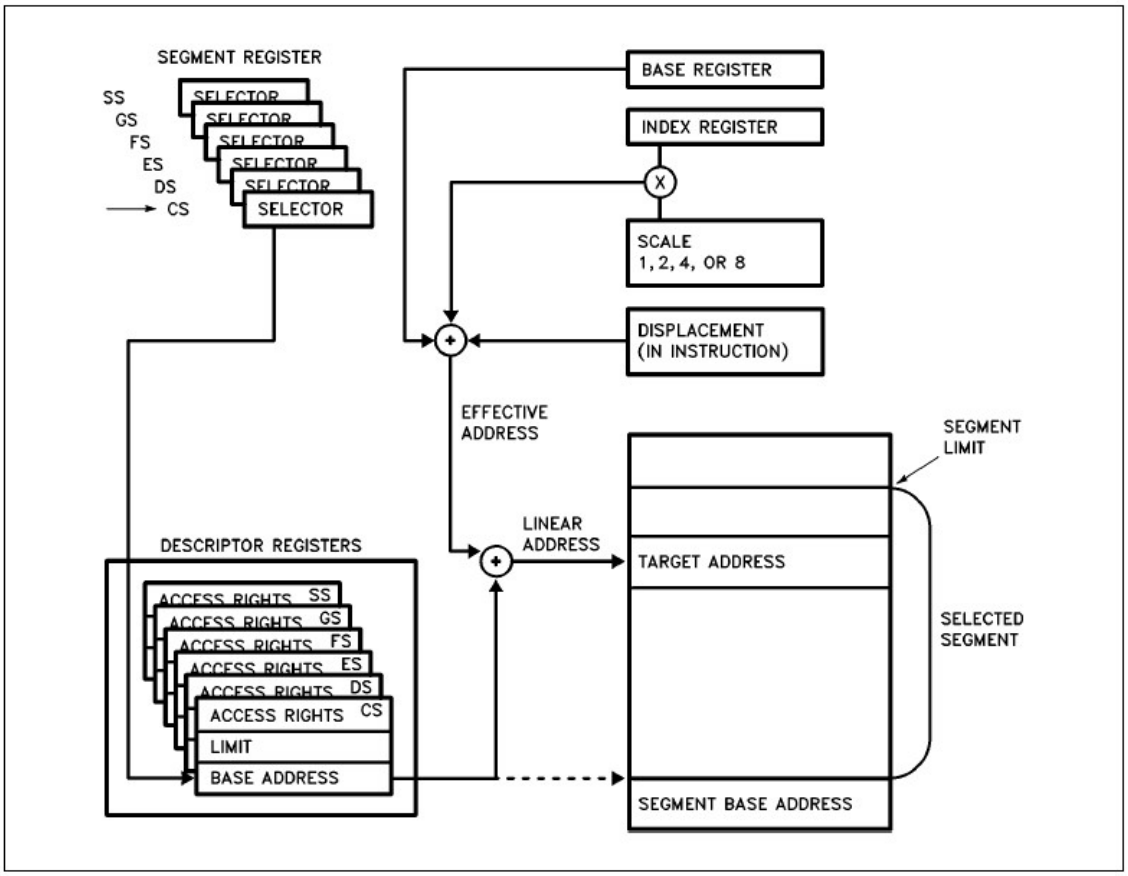


Fig. G.2.3 – Cálculo de la dirección física

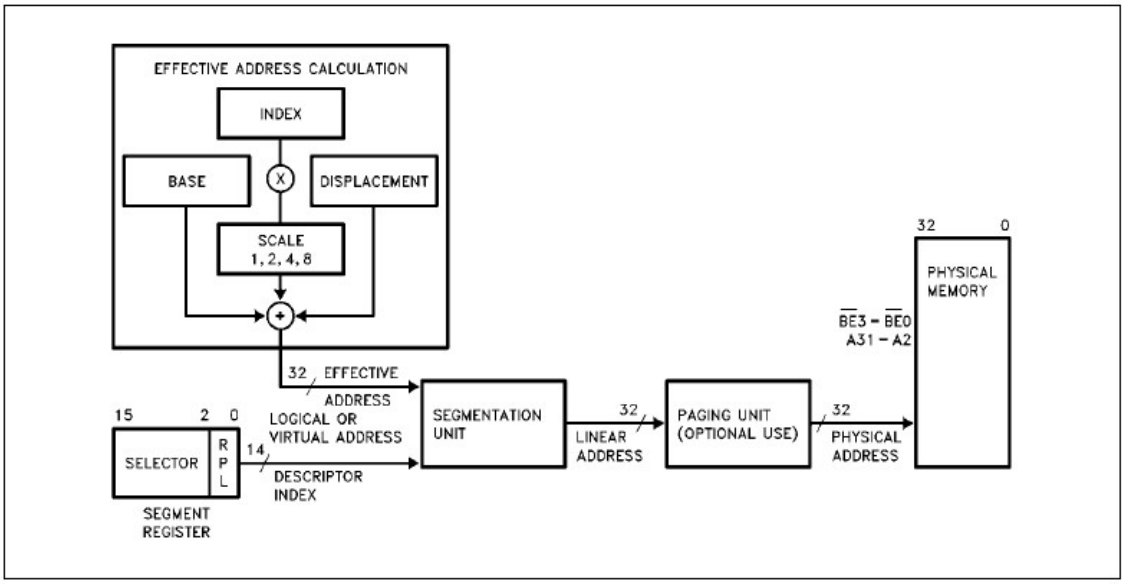


Fig. G.2.4 – Diagrama completo con segmentación y paginación

G.2.6 El sistema de interrupciones

Function	Interrupt Number	Instruction Which Can Cause Exception	Return Address Points to Faulting Instruction	Type
Divide Error	0	DIV, IDIV	YES	FAULT
Debug Exception	1	any instruction	YES	TRAP*
NMI Interrupt	2	INT 2 or NMI	NO	NMI
One Byte Interrupt	3	INT	NO	TRAP
Interrupt on Overflow	4	INTO	NO	TRAP
Array Bounds Check	5	BOUND	YES	FAULT
Invalid OP-Code	6	Any Illegal Instruction	YES	FAULT
Device Not Available	7	ESC, WAIT	YES	FAULT
Double Fault	8	Any Instruction That Can Generate an Exception		ABORT
Coprocessor Segment Overrun	9	ESC	NO	ABORT
Invalid TSS	10	JMP, CALL, IRET, INT	YES	FAULT
Segment Not Present	11	Segment Register Instructions	YES	FAULT
Stack Fault	12	Stack References	YES	FAULT
General Protection Fault	13	Any Memory Reference	YES	FAULT
Intel Reserved	15			
Page Fault	14	Any Memory Access or Code Fetch	YES	FAULT
Coprocessor Error	16	ESC, WAIT	YES	FAULT
Intel Reserved	17–31			
Two Byte Interrupt	0–255	INT n	NO	TRAP

Tabla G.2.2 - Asignación de vectores en el 80386

G.2.7. Acceso directo a memoria

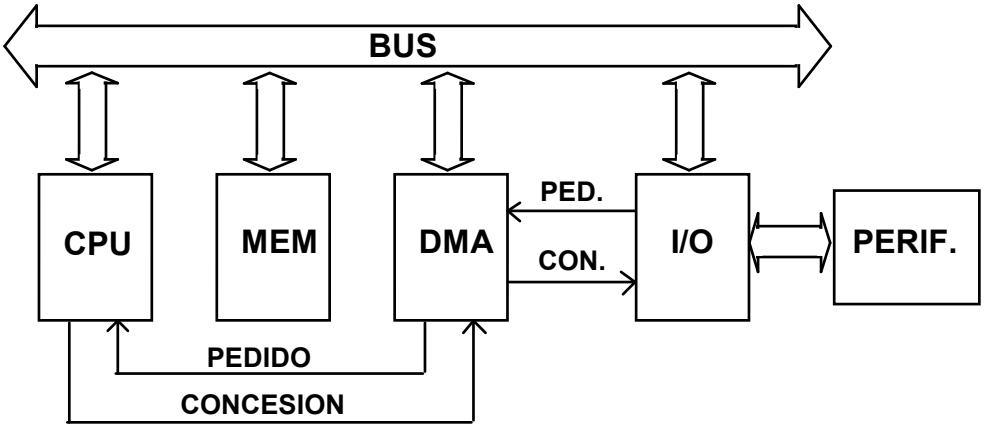


Fig. G.2.5 - Configuración circuital

G.3 EL PENTIUM

G.3.1 Estructura interna

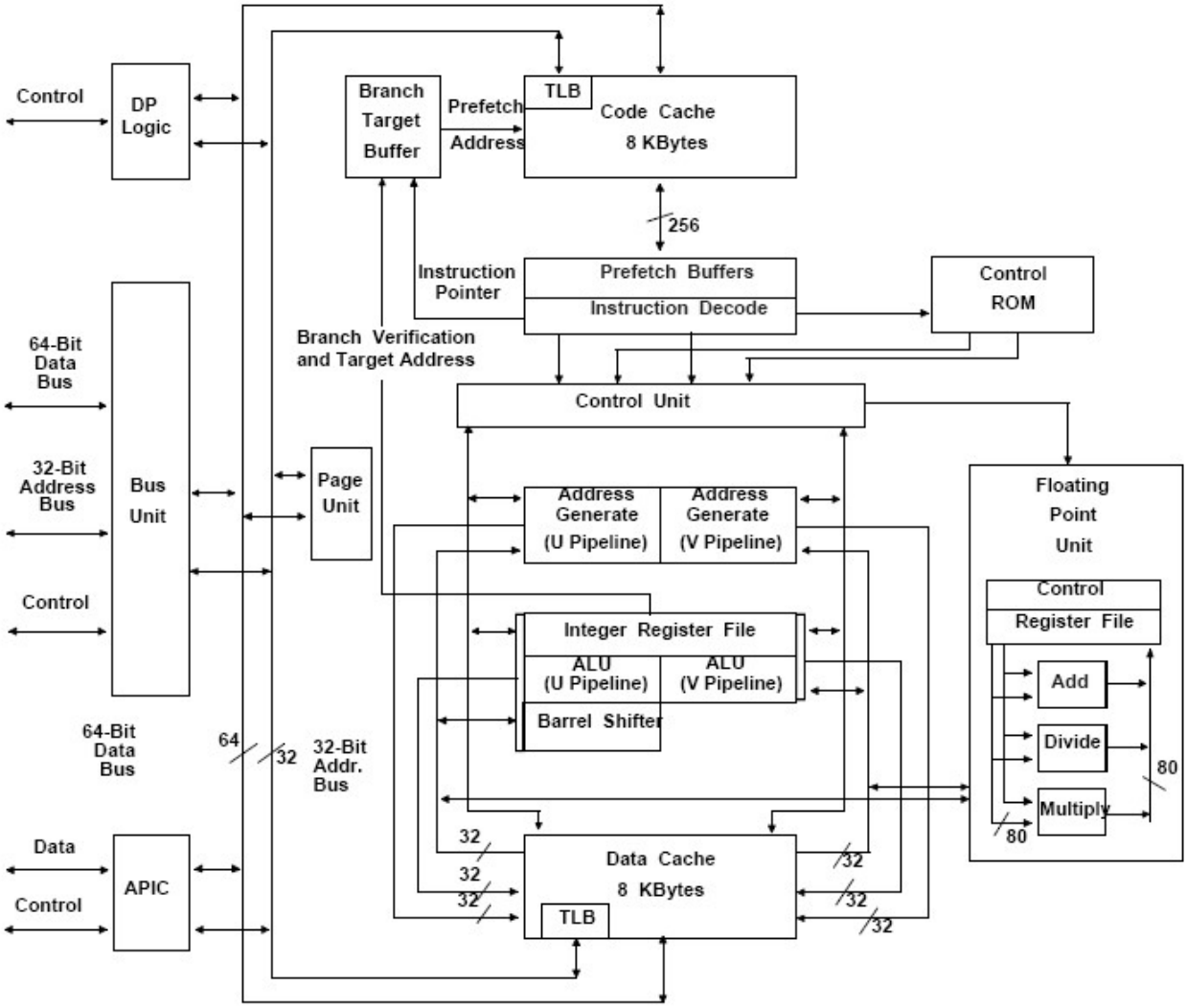


Figura G.3.1 - Diagrama funcional simplificado del Pentium

BIBLIOGRAFÍA

1. Sistemas Operativos Modernos – 3ª edición (2012) – Andrew S. Tanenbaum – PEARSON EDUCACION
2. Organización de Computadoras – Un enfoque estructurado – 5ª edición (2010) – Andrew S. Tanenbaum – PEARSON EDUCACION
3. The Scientist and Engineer's Guide to Digital Signal Processing - Steven W. Smith – http://www.DSPguide.com
4. Tratamiento de señales en tiempo discreto – 3ª edición (2011) – Oppenheim, Schafer, Buck – PRENTICE HALL
5. Curso Tratamiento digital de señal - Andoni Irizar Picón - TECNUN, Escuela Superior de Ingenieros de la Universidad de Navarra.
6. Redes de Ordenadores – 5ª edición (2013) - Andrew S. Tanenbaum y David J. Wetherall – PEARSON EDUCACION
7. 8086 – Technical manual - PDF - INTEL
8. 80386 – Technical manual - PDF - INTEL
9. 80387 – Technical manual - PDF - INTEL
10. 80386 – Programmer’s reference manual - PDF - INTEL
11. Pentium – Data sheet - PDF - INTEL
12. Athlon processors– Technical brief - PDF - AMD
13. Microprocesadores de 16 Bits - M Aumiaux - MASSON
14. ASM86 Language Reference Manual - INTEL CORPORATION